Network Working Group
Request for Comments: 3331
Category: Standards Track

K. Morneault
Cisco Systems
R. Dantu
NetRake
G. Sidebottom
Signatus Technologies
B. Bidulock
OpenSS7
J. Heitz
Lucent

September 2002

Signaling System 7 (SS7) Message Transfer Part 2 (MTP2) - User Adaptation Layer

Status of this Memo

Copyright Notice

Abstract

This document defines a protocol for the backhauling of Signaling System 7 Message Transfer Part 2 (SS7 MTP2) User signalling messages over IP using the Stream Control Transmission Protocol (SCTP). This protocol would be used between a Signalling Gateway (SG) and Media Gateway Controller (MGC). It is assumed that the SG receives SS7 signalling over a standard SS7 interface using the SS7 Message Transfer Part (MTP) to provide transport. The Signalling Gateway would act as a Signalling Link Terminal.

Table of Contents

1. Introduction

This document defines a protocol for the backhauling of SS7 [1] MTP2 User [2] [3] [4] (i.e. MTP3) signalling messages over IP using the Stream Control Transmission Protocol (SCTP) [8]. This protocol would be used between a Signalling Gateway (SG) and Media Gateway Controller (MGC).

1.1 Scope

There is a need for Switched Circuit Network (SCN) signalling protocol delivery from a Signalling Gateway (SG) to a Media Gateway Controller (MGC) [9]. The delivery mechanism addresses the following objectives:

* Support for MTP Level 2 / MTP Level 3 interface boundary
* Support for communication between Layer Management modules on SG and MGC
* Support for management of SCTP active associations between the SG and MGC

The SG will terminate up to MTP Level 2 and the MGC will terminate MTP Level 3 and above. In other words, the SG will transport MTP Level 3 messages over an IP network to a MGC.

1.2 Terminology

Application Server (AS) - A logical entity serving a specific application instance. An example of an Application Server is a MGC handling the MTP Level 3 and call processing for SS7 links terminated by the Signalling Gateways. Practically speaking, an AS is modeled at the SG as an ordered list of one or more related Application Server Processes (e.g., primary, secondary, tertiary, ...).

Application Server Process (ASP) - A process instance of an Application Server. Examples of Application Server Processes are active or standby MGC instances.

Association - An association refers to a SCTP association. The association will provide the transport for the delivery of protocol data units for one or more interfaces.

Backhaul - Refers to the transport of signalling from the point of interface for the associated data stream (i.e., SG function in the MGU) back to the point of call processing (i.e., the MGCU), if this is not local [9].

Fail-over - The capability to reroute signalling traffic as required to an alternate Application Server Process within an Application Server in the event of failure or unavailability of a currently used Application Server Process. Fail-back MAY apply upon the return to service of a previously unavailable Application Server Process.

Host - The computing platform that the ASP process is running on.

Interface - For the purposes of this document, an interface is a SS7 signalling link.

Interface Identifier - The Interface Identifier identifies the physical interface at the SG for which the signalling messages are sent/received. The format of the Interface Identifier parameter can be text or integer, the values of which are assigned according to network operator policy. The values used are of local significance only, coordinated between the SG and ASP.

Layer Management - Layer Management is a nodal function in an SG or ASP that handles the inputs and outputs between the M2UA layer and a local management entity.

Link Key - The link key is a locally unique (between ASP and SG) value that identifies a registration request for a particular Signalling Data Link and Signalling Terminal pair.

MTP - The Message Transfer Part of the SS7 protocol

MTP2 - MTP Level 2, the signalling data link layer of SS7

MTP3 - MTP Level 3, the signalling network layer of SS7

MTP2-User - A protocol that uses the services of MTP Level 2 (i.e. MTP3).

Network Byte Order: Most significant byte first, a.k.a Big Endian.

Signalling Data Link - An SDL refers to a specific communications facility that connects two Signalling Link Terminals.

Signalling Gateway (SG) - An SG is a signalling agent at the edge of the IP network. An SG appears to the SS7 as one or more Signalling Link Terminals that are connected to one or more Signalling Data Links in the SS7 network. An SG contains a set of one or more unique Signalling Gateway Processes, on which one or more is normally actively processing traffic. Where an SG contains more than one SGP, the SG is a logical entity.

Signalling Gateway Process (SGP) - A process instance that uses M2UA to communicate to and from a Signalling Link Terminal. It serves as an active, backup or load-sharing process of a Signalling Gateway.

Signalling Link Terminal (SLT) - Refers to the means of performing all of the functions defined at MTP level 2 regardless of their implementation [2,3].

Stream - A stream refers to an SCTP stream; a unidirectional logical channel established from one SCTP endpoint to another associated SCTP endpoint, within which all user messages are delivered in-sequence except for those submitted to the unordered delivery service.

1.3 M2UA Overview

The framework architecture that has been defined for SCN signalling transport over IP [9] uses two components: a signalling common transport protocol and an adaptation module to support the services expected by a particular SCN signalling protocol from its underlying protocol layer.

Within this framework architecture, this document defines a SCN adaptation module that is suitable for the transport of SS7 MTP2 User messages. The only SS7 MTP2 User is MTP3. The M2UA uses the services of the Stream Control Transmission Protocol [8] as the underlying reliable signalling common transport protocol.

In a Signalling Gateway, it is expected that the SS7 MTP2-User signalling is transmitted and received from the PSTN over a standard SS7 network interface, using the SS7 Message Transfer Part Level 1 and Level 2 [2,3,4] to provide reliable transport of the MTP3-User signalling messages to and from an SS7 Signalling End Point (SEP) or Signalling Transfer Point (STP). The SG then provides an interworking of transport functions with the IP transport, in order to transfer the MTP2-User signalling messages to and from an Application Server Process where the peer MTP2-User protocol layer exists.

1.3.1 Example - SG to MGC

In a Signalling Gateway, it is expected that the SS7 signalling is received over a standard SS7 network termination, using the SS7 Message Transfer Part (MTP) to provide transport of SS7 signalling messages to and from an SS7 Signalling End Point (SEP) or SS7 Signalling Transfer Point (STP). In other words, the SG acts as a Signalling Link Terminal (SLT) [2,3]. The SG then provides an interworking of transport functions with IP Signalling Transport, in order to transport the MTP3 signalling messages to the MGC where the peer MTP3 protocol layer exists, as shown below:

```
    ******    SS7    ******      IP     *******
    *SEP *-----------* SG *-------------* MGC *
    ******           ******             *******

    +----+                              +----+
    |S7UP|                              |S7UP|
    +----+                              +----+
    |MTP +                              |MTP |
    | L3 |          (NIF)               |L3  |
    +----+         +----+----+          +----+
    |MTP |         |MTP |M2UA|          |M2UA|
    |    |         |    +----+          +----+
    |L2  |         |L2  |SCTP|          |SCTP|
    |L1  |         |L1  +----+          +----+
    |    |         |    |IP  |          |IP  |
    +----+         +---------+          +----+

    NIF  - Nodal Interworking Function
    SEP  - SS7 Signalling Endpoint
    IP   - Internet Protocol
    SCTP - Stream Control Transmission Protocol (Reference [8])
```

Figure 1 M2UA in the SG to MGC Application

Note: STPs MAY be present in the SS7 path between the SEP and the SG.

It is recommended that the M2UA use the services of the Stream Control Transmission Protocol (SCTP) [8] as the underlying reliable common signalling transport protocol. The use of SCTP provides the following features:

- explicit packet-oriented delivery (not stream-oriented)
- sequenced delivery of user messages within multiple streams, with an option for order-of-arrival delivery of individual user messages,
- optional multiplexing of user messages into SCTP datagrams,

- network-level fault tolerance through the support of multi-homing at either or both ends of an association,
- resistance to flooding and masquerade attacks, and
- data segmentation to conform to discovered path MTU size

There are scenarios without redundancy requirements and scenarios in which redundancy is supported below the transport layer. In these cases, the SCTP functions above MAY NOT be a requirement and TCP can be used as the underlying common transport protocol.

1.3.2 ASP Fail-over Model and Terminology

The M2UA layer supports ASP fail-over functions in order to support a high availability of call and transaction processing capability. All MTP2-User messages incoming to a SGP from the SS7 network are assigned to the unique Application Server, based on the Interface Identifier of the message.

The M2UA layer supports a n+k redundancy model (active-standby, load sharing, broadcast) where n is the minimum number of redundant ASPs required to handle traffic and k ASPs are available to take over for a failed or unavailable ASP. Note that 1+1 active/standby redundancy is a subset of this model. A simplex 1+0 model is also supported as a subset, with no ASP redundancy.

1.3.3 Client/Server Model

It is recommended that the SGP and ASP be able to support both client and server operation. The peer endpoints using M2UA SHOULD be configured so that one always takes on the role of client and the other the role of server for initiating SCTP associations. The default orientation would be for the SGP to take on the role of server while the ASP is the client. In this case, ASPs SHOULD initiate the SCTP association to the SGP.

The SCTP and TCP Registered User Port Number Assignment for M2UA is 2904.

1.4 Services Provided by the M2UA Adaptation Layer

The SS7 MTP3/MTP2(MTP2-User) interface is retained at the termination point in the IP network, so that the M2UA protocol layer is required to provide the equivalent set of services to its users as provided by the MTP Level 2 to MTP Level 3.

1.4.1 Support for MTP Level 2 / MTP Level 3 interface boundary

M2UA supports a MTP Level 2 / MTP Level 3 interface boundary that enables a seamless, or as seamless as possible, operation of the MTP2-User peers in the SS7 and IP domains. An example of the primitives that need to be supported can be found in [10].

1.4.2 Support for communication between Layer Management modules on SG and MGC

The M2UA layer needs to provide some messages that will facilitate communication between Layer Management modules on the SG and MGC. To facilitate reporting of errors that arise because of the backhauling MTP Level 3 scenario, the following primitive is defined:

M-ERROR

The M-ERROR message is used to indicate an error with a received M2UA message (e.g., an interface identifier value is not known to the SG).

1.4.3 Support for management of active associations between SG and MGC

The M2UA layer on the SG keeps the state of the configured ASPs. A set of primitives between M2UA layer and the Layer Management are defined below to help the Layer Management manage the association(s) between the SG and the MGC. The M2UA layer can be instructed by the Layer Management to establish a SCTP association to a peer M2UA node. This procedure can be achieved using the M-SCTP ESTABLISH primitive.

M-SCTP_ESTABLISH

The M-SCTP_ESTABLISH primitive is used to request, indicate and confirm the establishment of a SCTP association to a peer M2UA node.

M-SCTP_RELEASE

The M-SCTP_RELEASE primitives are used to request, indicate, and confirm the release of a SCTP association to a peer M2UA node.

The M2UA layer MAY also need to inform the status of the SCTP association(s) to the Layer Management. This can be achieved using the following primitive.

M-SCTP_STATUS

The M-SCTP_STATUS primitive is used to request and indicate the status of underlying SCTP association(s).

The Layer Management MAY need to inform the M2UA layer of an AS/ASP status (i.e., failure, active, etc.), so that messages can be exchanged between M2UA layer peers to stop traffic to the local M2UA user. This can be achieved using the following primitive.

M-ASP_STATUS

The ASP status is stored inside the M2UA layer on both the SG and MGC sides. The M-ASP_STATUS primitive can be used by Layer Management to request the status of the Application Server Process from the M2UA layer. This primitive can also be used to indicate the status of the Application Server Process.

M-ASP_MODIFY

The M-ASP_MODIFY primitive can be used by Layer Management to modify the status of the Application Server Process. In other words, the Layer Management on the ASP side uses this primitive to initiate the ASPM procedures.

M-AS_STATUS

The M-AS_STATUS primitive can be used by Layer Management to request the status of the Application Server. This primitive can also be used to indicate the status of the Application Server.

1.5 Functions Provided by the M2UA Layer

1.5.1 Mapping

The M2UA layer MUST maintain a map of an Interface ID to a physical interface on the Signalling Gateway. A physical interface would be a V.35 line, T1 line/time slot, E1 line/time slot, etc. The M2UA layer MUST also maintain a map of the Interface Identifier to SCTP association and to the related stream within the association.

The SGP maps an Interface Identifier to an SCTP association/stream only when an ASP sends an ASP Active message for a particular Interface Identifier. It must be noted, however, that this mapping is dynamic and could change at any time due to a change of ASP state. This mapping could even temporarily be invalid, for example during fail-over of one ASP to another. Therefore, the SGP MUST maintain the states of AS/ASP and reference them during the routing of any messages to an AS/ASP.

Note that only one SGP SHOULD provide Signalling Link Terminal services to an SS7 link. Therefore, within an SG, an Application Server SHOULD be active for only one SGP at any given point in time.

An example of the logical view of the relationship between an SS7 link, Interface Identifier, AS and ASP in an SGP is shown below:

```
               /-------------------------------------------------+
              /   /----------------------------------------------|--+
             /   /                                               v  |
            /   /    +----+             act+-----+   +-------+ -+--+|-+-
   SS7 link1-------->|IID |-+          +-->| ASP |-->| Assoc |      v
              /      +----+ |  +----+  |   +-----+   +-------+ -+--+--+-
             /              +->| AS |--+                        Streams
            /        +----+ |  +----+   stb+-----+
   SS7 link2-------->|IID |-+              | ASP |
                     +----+                +-----+
```

where IID = Interface Identifier

A SGP MAY support more than one AS. An AS MAY support more than one Interface Identifier.

1.5.2 Support for the management of SCTP associations between the SGPs and ASPs

The M2UA layer at the SG maintains the availability state of all configured ASPs, in order to manage the SCTP associations and the traffic between the SG and ASPs. As well, the active/inactive state of remote ASP(s) are also maintained. The Active ASP(s) are the one(s) currently receiving traffic from the SG.

The M2UA layer MAY be instructed by local management to establish an SCTP association to a peer M2UA node. This can be achieved using the M-SCTP_ESTABLISH primitive to request, indicate and confirm the establishment of an SCTP association with a peer M2UA node.

The M2UA layer MAY also need to inform local management of the status of the underlying SCTP associations using the M-SCTP_STATUS request and the indication primitive. For example, the M2UA MAY inform local management of the reason for the release of an SCTP association, determined either locally within the M2UA layer or by a primitive from the SCTP.

Also the M2UA layer may need to inform the local management of the change in status of an ASP or AS. This may be achieved using the M-ASP STATUS request or M-AS_STATUS request primitives.

1.5.3 Status of ASPs

The M2UA layer on the SG MUST maintain the state of the ASPs it is supporting. The state of an ASP changes because of the reception of peer-to-peer messages (ASPM messages as described in Section 3.3.2) or the reception of indications from the local SCTP association. The ASP state transition procedures are described in Section 4.3.1.

At a SGP, an Application Server list MAY contain active and inactive ASPs to support ASP fail-over procedures. When, for example, both a primary and a backup ASP are available, the M2UA peer protocol is required to control which ASP is currently active. The ordered list of ASPs within a logical Application Server is kept updated in the SGP to reflect the active Application Server Process.

Also the M2UA layer MAY need to inform the local management of the change in status of an ASP or AS. This can be achieved using the M-ASP_STATUS or M-AS_STATUS primitives.

1.5.4 SCTP Specifics

1.5.4.1 SCTP Stream Management

SCTP allows a user specified number of streams to be opened during initialization of the association. It is the responsibility of the M2UA layer to ensure proper management of these streams. Because of the unidirectional nature of streams, a M2UA layer is not aware of the stream information from its peer M2UA layer. For this reason, the Interface Identifier is in the M2UA message header.

The use of SCTP streams within M2UA is recommended in order to minimize transmission and buffering delay, thereby, improving the overall performance and reliability of the signalling elements. A separate SCTP stream can be used for each SS7 link. Or, an implementation may choose to split the SS7 link across several streams based on SLS. This method may be of particular interest for high speed SS7 links (MTP3b) since high speed links have a 24-bit sequence number and the stream sequence number is 16-bits.

SCTP Stream ’0’ SHOULD NOT be used for MTP2 User Adaptation (MAUP) messages (see Section 3) since stream ’0’ SHOULD only be used for ASP Management (ASPM) messages (see Section 4.3.3).

1.5.5 Seamless SS7 Network Management Interworking

The M2UA layer on the SGP SHOULD pass an indication of unavailability of the M2UA-User (MTP3) to the local Layer Management, if the currently active ASP moves from the ACTIVE state. The actions taken by M2UA on the SGP with regards to MTP Level 2 should be in accordance with the appropriate MTP specifications.

1.5.6 Flow Control / Congestion

It is possible for the M2UA layer to be informed of the IP network congestion onset and abatement by means of an implementation dependent function (i.e. an indication from the SCTP). The handling of this congestion indication by M2UA is implementation dependent. However, the actions taken by the SG should be in accordance with the appropriate MTP specification and should enable SS7 functionality (e.g. flow control) to be correctly maintained.

1.5.7 Audit of SS7 Link State

After a fail-over of one ASP to another ASP, it may be necessary for the M2UA on the ASP to audit the current SS7 link state to ensure consistency. The M2UA on the SGP would respond to the audit request with information regarding the current state of the SS7 link (i.e. in-service, out-of-service, congestion state, LPO/RPO state).

1.6 Definition of the M2UA Boundaries

1.6.1 Definition of the M2UA / MTP Level 3 boundary

DATA
ESTABLISH
RELEASE
STATE
DATA RETRIEVAL
DATA RETRIEVAL COMPLETE

1.6.2 Definition of the M2UA / MTP Level 2 boundary

DATA
ESTABLISH
RELEASE
STATE
DATA RETRIEVAL
DATA RETRIEVAL COMPLETE

1.6.3 Definition of the Lower Layer Boundary between M2UA and SCTP

The upper layer and layer management primitives provided by SCTP are provided in Reference [8] Section 10.

1.6.4 Definition of Layer Management / M2UA Boundary

M-SCTP_ESTABLISH request
Direction: LM -> M2UA
Purpose: LM requests ASP to establish an SCTP association with an SGP.

M-SCTP_ESTABLISH confirm
Direction: M2UA -> LM
Purpose: ASP confirms to LM that it has established an SCTP association with an SGP.

M-SCTP_ESTABLISH indication
Direction: M2UA -> LM
Purpose: SGP informs LM that an ASP has established an SCTP association.

M-SCTP_RELEASE request
Direction: LM -> M2UA
Purpose: LM requests ASP to release an SCTP association with SGP.

M-SCTP_RELEASE confirm
Direction: M2UA -> LM
Purpose: ASP confirms to LM that it has released SCTP association with SGP.

M-SCTP_RELEASE indication
Direction: M2UA -> LM
Purpose: SGP informs LM that ASP has released an SCTP association.

M-SCTP_RESTART indication
Direction: M2UA -> LM
Purpose: M2UA informs LM that a SCTP Restart indication has been received.

M-SCTP_STATUS request
Direction: LM -> M2UA
Purpose: LM requests M2UA to report status of SCTP association.

M-SCTP_STATUS indication
Direction: M2UA -> LM
Purpose: M2UA reports status of SCTP association.

M-ASP_STATUS request
Direction: LM -> M2UA
Purpose: LM requests SGP to report status of remote ASP.

M-ASP_STATUS indication
Direction: M2UA -> LM
Purpose: SGP reports status of remote ASP.

M-AS_STATUS request
Direction: LM -> M2UA
Purpose: LM requests SG to report status of AS.

M-AS_STATUS indication
Direction: M2UA -> LM
Purpose: SG reports status of AS.

M-NOTIFY indication
Direction: M2UA -> LM
Purpose: ASP reports that it has received a NOTIFY message from its peer.

M-ERROR indication
Direction: M2UA -> LM
Purpose: ASP or SGP reports that it has received an ERROR message from its peer.

M-ASP_UP request
Direction: LM -> M2UA
Purpose: LM requests ASP to start its operation and send an ASP UP message to the SGP.

M-ASP_UP confirm
Direction: M2UA -> LM
Purpose: ASP reports that it has received an ASP UP Acknowledgment message from the SGP.

M-ASP_DOWN request
Direction: LM -> M2UA
Purpose: LM requests ASP to stop its operation and send an ASP DOWN message to the SGP.

M-ASP_DOWN confirm
Direction: M2UA -> LM
Purpose: ASP reports that is has received an ASP DOWN Acknowledgment message from the SGP.

```
   M-ASP_ACTIVE request
   Direction: LM -> M2UA
   Purpose: LM requests ASP to send an ASP ACTIVE message to the SGP.

   M-ASP_ACTIVE confirm
   Direction: M2UA -> LM
   Purpose: ASP reports that is has received an ASP ACTIVE
            Acknowledgment message from the SGP.

   M-ASP_INACTIVE request
   Direction: LM -> M2UA
   Purpose: LM requests ASP to send an ASP INACTIVE message to the SGP.

   M-ASP_INACTIVE confirm
   Direction: M2UA -> LM
   Purpose: ASP reports that is has received an ASP INACTIVE
            Acknowledgment message from the SGP.

   M-LINK_KEY_REG Request
   Direction:  LM -> M2UA
   Purpose: LM requests ASP to register Link Key with SG by sending REG
            REQ message.

   M-LINK_KEY_REG Confirm
   Direction:   M2UA -> LM
   Purpose: ASP reports to LM that it has successfully received a REG
            RSP message from SG.

   M-LINK_KEY_REG Indication
   Direction:  M2UA -> LM
   Purpose:  SG reports to LM that it has successfully processed an
             incoming REG REQ message from ASP.

   M-LINK_KEY_DEREG Request
   Direction:  LM -> M2UA
   Purpose:  LM requests ASP to de-register Link Key with SG by sending
             DEREG REQ message.

   M-LINK_KEY_DEREG Confirm
   Direction:  M2UA -> LM
   Purpose:  ASP reports to LM that it has successfully received a
             DEREG RSP message from SG.

   M-LINK_KEY_DEREG  Indication
   Direction:  M2UA -> LM
   Purpose:  SG reports to LM that it has successfully processed an
             incoming DEREG REQ message from ASP.
```

2.0 Conventions

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, NOT RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

3.0 Protocol Elements

This section describes the format of various messages used in this protocol.

3.1 Common Message Header

The protocol messages for MTP2-User Adaptation require a message structure that contains a version, message class, message type, message length, and message contents. This message header is common among all signalling protocol adaptation layers:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |     Spare     | Message Class | Message Type  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Message Length                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2 Common Message Header

All fields in an M2UA message MUST be transmitted in the network byte order, unless otherwise stated.

3.1.1 Version

The version field contains the version of the M2UA adaptation layer. The supported versions are:

Value	Version
1	Release 1.0

3.1.2 Spare

The Spare field is 8-bits. It SHOULD be set to all ’0’s by the sender and ignored by the receiver.

3.1.3 Message Class

The following List contains the valid Message Classes:

Message Class: 8 bits (unsigned integer)

0	Management (MGMT) Message [IUA/M2UA/M3UA/SUA]
1	Transfer Messages [M3UA]
2	SS7 Signalling Network Management (SSNM) Messages [M3UA/SUA]
3	ASP State Maintenance (ASPSM) Messages [IUA/M2UA/M3UA/SUA]
4	ASP Traffic Maintenance (ASPTM) Messages [IUA/M2UA/M3UA/SUA]
5	Q.921/Q.931 Boundary Primitives Transport (QPTM) Messages [IUA]
6	MTP2 User Adaptation (MAUP) Messages [M2UA]
7	Connectionless Messages [SUA]
8	Connection-Oriented Messages [SUA]
9	Routing Key Management (RKM) Messages (M3UA)
10	Interface Identifier Management (IIM) Messages (M2UA)
11 to 127	Reserved by the IETF
128 to 255	Reserved for IETF-Defined Message Class extensions

3.1.4 Message Type

The following List contains the Message Types for the valid Message Classes:

MTP2 User Adaptation (MAUP) Messages

0	Reserved
1	Data
2	Establish Request
3	Establish Confirm
4	Release Request
5	Release Confirm
6	Release Indication
7	State Request
8	State Confirm
9	State Indication
10	Data Retrieval Request
11	Data Retrieval Confirm
12	Data Retrieval Indication
13	Data Retrieval Complete Indication
14	Congestion Indication
15	Data Acknowledge
16 to 127	Reserved by the IETF
128 to 255	Reserved for IETF-Defined MAUP extensions

Application Server Process State Maintenance (ASPSM) messages

```
     0      Reserved
     1      ASP Up (UP)
     2      ASP Down (DOWN)
     3      Heartbeat (BEAT)
     4      ASP Up Ack (UP ACK)
     5      ASP Down Ack (DOWN ACK)
     6      Heartbeat Ack (BEAT ACK)
  7 to 127  Reserved by the IETF
128 to 255  Reserved for IETF-Defined ASPSM extensions
```

Application Server Process Traffic Maintenance (ASPTM) messages

```
     0      Reserved
     1      ASP Active (ACTIVE)
     2      ASP Inactive (INACTIVE)
     3      ASP Active Ack (ACTIVE ACK)
     4      ASP Inactive Ack (INACTIVE ACK)
  5 to 127  Reserved by the IETF
128 to 255  Reserved for IETF-Defined ASPTM extensions
```

Management (MGMT) Messages

```
     0      Error (ERR)
     1      Notify (NTFY)
  2 to 127  Reserved by the IETF
128 to 255  Reserved for IETF-Defined MGMT extensions
```

Interface Identifier Management (IIM) Messages

```
     0        Reserved
     1        Registration Request (REG REQ)
     2        Registration Response (REG RSP)
     3        Deregistration Request (DEREG REQ)
     4        Deregistration Response (DEREG RSP)
  5 to 127    Reserved by the IETF
128 to 255    Reserved for IETF-Defined IIM extensions
```

3.1.5 Message Length

The Message Length defines the length of the message in octets, including the header. The Message Length MUST include parameter padding bytes, if any. The Message Length MUST NOT be longer than a MTP3 message [2,3,4,5] plus the length of the common and M2UA message headers.

3.1.6 Variable-Length Parameter Format

M2UA messages consist of a Common Header followed by zero or more variable-length parameters, as defined by the message type. The variable-length parameters contained in a message are defined in a Tag-Length-Value format as shown below.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Parameter Tag        |       Parameter Length        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                       Parameter Value                         /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Mandatory parameters MUST be placed before optional parameters in a message.

Parameter Tag: 16 bits (unsigned integer)

The Type field is a 16 bit identifier of the type of parameter. It takes a value of 0 to 65534. The common parameters used by the adaptation layers are in the range of 0x00 to 0xff. The M2UA specific parameters have Tags in the range 0x300 to 0x3ff.

The common parameter tags (used by all User Adaptation layers) that M2UA uses are defined below:

Parameter Value	Parameter Name
0 (0x00)	Reserved
1 (0x01)	Interface Identifier (Integer)
2 (0x02)	Unused
3 (0x03)	Interface Identifier (Text)
4 (0x04)	Info String
5 (0x05)	Unused
6 (0x06)	Unused
7 (0x07)	Diagnostic Information
8 (0x08)	Interface Identifier (Integer Range)
9 (0x09)	Heartbeat Data
10 (0x0a)	Unused
11 (0x0b)	Traffic Mode Type
12 (0x0c)	Error Code
13 (0x0d)	Status Type/Information
14 (0x0e)	Unused
15 (0x0f)	Unused
16 (0x10)	Unused
17 (0x11)	ASP Identifier
18 (0x12)	Unused
19 (0x13)	Correlation Id
18-255	Reserved

The M2UA specific parameter Tags defined are as follows:

Parameter Value	Parameter Name
768 (0x0300)	Protocol Data 1
769 (0x0301)	Protocol Data 2 (TTC)
770 (0x0302)	State Request
771 (0x0303)	State Event
772 (0x0304)	Congestion Status
773 (0x0305)	Discard Status
774 (0x0306)	Action
775 (0x0307)	Sequence Number
776 (0x0308)	Retrieval Result
777 (0x0309)	Link Key
778 (0x030a)	Local-LK-Identifier
779 (0x030b)	Signalling Data Terminal (SDT) Identifier
780 (0x030c)	Signalling Data Link (SDL) Identifier
781 (0x030d)	Registration Result
782 (0x030e)	Registration Status
783 (0x030f)	De-Registration Result
784 (0x0310)	De-Registration Status

Parameter Length: 16 bits (unsigned integer)

The Parameter Length field contains the size of the parameter in bytes, including the Parameter Tag, Parameter Length, and Parameter Value fields. Thus, a parameter with a zero-length Parameter Value field would have a Length field of 4. The Parameter Length does not include any padding bytes.

Parameter Value: variable-length.

The Parameter Value field contains the actual information to be transferred in the parameter.

The total length of a parameter (including Tag, Parameter Length and Value fields) MUST be a multiple of 4 bytes. If the length of the parameter is not a multiple of 4 bytes, the sender pads the Parameter at the end (i.e., after the Parameter Value field) with all zero bytes. The length of the padding is NOT included in the parameter length field. A sender MUST NOT pad with more than 3 bytes. The receiver MUST ignore the padding bytes.

3.2 M2UA Message Header

In addition to the common message header, there will be a M2UA specific message header. The M2UA specific message header will immediately follow the common message header, but will only be used with MAUP messages.

This message header will contain the Interface Identifier. The Interface Identifier identifies the physical interface at the SG for which the signalling messages are sent/received. The format of the Interface Identifier parameter can be text or integer, the values of which are assigned according to network operator policy. The values used are of local significance only, coordinated between the SG and ASP.

The integer formatted Interface Identifier MUST be supported. The text formatted Interface Identifier MAY optionally be supported.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0x1)           |           Length=8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Interface Identifier (integer)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3 M2UA Message Header (Integer-based Interface Identifier)

The Tag value for the Integer-based Interface Identifier is 0x1. The length is always set to a value of 8.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0x3)           |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                   Interface Identifier (text)                 /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4 M2UA Message Header (Text-based Interface Identifier)

The Tag value for the Text-based Interface Identifier is 0x3. The encoding of the Identifier is ANSI X3.4-1986 [7]. The maximum string length of the text-based Interface Identifier is 255 octets. The tag length is equal to the string length of the Interface Identifier name plus four bytes for the Tag and Length fields.

3.3 M2UA Messages

The following section defines the messages and parameter contents. The M2UA messages will use the common message header (Figure 2) and the M2UA message header (Figure 3 and Figure 4).

3.3.1 MTP2 User Adaptation Messages

3.3.1.1 Data

The Data message contains an SS7 MTP2-User Protocol Data Unit (PDU). The Data message contains the following parameter:

    Protocol Data (mandatory)
    Correlation Id (optional)

The format for the Data Message parameters is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x300)           |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          Protocol Data                        /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x13)            |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Correlation Id                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Protocol Data field contains the MTP2-User application message in network byte order starting with the Signalling Information Octet (SIO). The Correlation Id parameter uniquely identifies the MSU carried in the Protocol Data within an AS. This Correlation Id parameter is assigned by the sending M2UA. The purpose of the Correlation Id is to permit the newly active ASP to synchronize its processing of the traffic in each ordered stream with other ASPs in the broadcast group.

The format for a Data Message with TTC PDU parameters is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x301)           |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                     TTC Protocol Data                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag (0x13)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Correlation Id                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Protocol Data field contains the MTP2-User application message in network byte order starting with the Length Indicator (LI) octet. The Japanese TTC variant uses the spare bits of the LI octet for priority.

The length of the Protocol Data and TTC Protocol Data MUST NOT exceed the length of a MTP2-User application message [2,3,5].

3.3.1.2 Data Acknowledge Message

The Data Acknowledge message contains the Correlation Id of the Data message that the sending M2UA is acknowledging as successfully processed to the peer M2UA.

The Data Acknowledge message contains the following parameter:

```
   Correlation Id       Mandatory
```

The following format MUST be used for the Data Ack Message:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag (0x13)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Correlation Id                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Correlation Id parameter of the Data message and the Data Ack message provide a mechanism, for those SG implementations capable of taking advantage of them, to obtain an acknowledgment that the MSU has been transferred to the M2UA peer before acknowledging the MSU to

the SS7 peer, removing the risk of losing messages due to association failure or SCTP congestion.

The Data Ack message MUST be sent if a Correlation Id parameter is received from the peer. Otherwise, the Data Ack message MUST NOT be sent.

If the Data Acknowledge is not sent for Correlation Id(s) or is sent with Invalid Correlation Id(s), the SS7 link will eventually fail due to lack of MTP Level 2 acknowledgments of the SS7 peer's MSUs.

3.3.1.3 Establish (Request, Confirmation)

The Establish Request message is used to establish the SS7 link or to indicate that the channel has been established. The MGC controls the state of the SS7 link. When the MGC desires the SS7 link to be in-service, it will send the Establish Request message. Note that the SGP MAY already have the SS7 link established at its layer. If so, upon receipt of an Establish Request, the SGP takes no action except to send an Establish Confirm.

When the MGC sends an M2UA Establish Request message, the MGC MAY start a timer. This timer would be stopped upon receipt of an M2UA Establish Confirm. If the timer expires, the MGC would resend the M2UA Establish Request message and restart the timer. In other words, the MGC MAY continue to request the establishment of the data link on a periodic basis until the desired state is achieved or some other action is taken (notify the Management Layer).

The mode (Normal or Emergency) for bringing the SS7 link in service is defaulted to Normal. The State Request (described in Section 3.3.1.5 below) can be used to change the mode to Emergency.

3.3.1.4 Release (Request, Indication, Confirmation)

This Release Request message is used to release the channel. The Release Confirm and Indication messages are used to indicate that the channel has been released.

3.3.1.5 State Request

The State Request message can be sent from a MGC to cause an action on a particular SS7 link supported by the Signalling Gateway Process. The SGP sends a State Confirm to the MGC if the action has been successfully completed. The State Confirm reflects that state value received in the State Request message.

The State Request message contains the following parameter:

State (mandatory)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x302)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             State                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The valid values for State are shown in the following table.

Define	Value	Description
STATUS_LPO_SET	0x0	Request local processor outage
STATUS_LPO_CLEAR	0x1	Request local processor outage recovered
STATUS_EMER_SET	0x2	Request emergency alignment
STATUS_EMER_CLEAR	0x3	Request normal alignment (cancel emergency)
STATUS_FLUSH_BUFFERS	0x4	Flush or clear receive, transmit and retransmit queues
STATUS_CONTINUE	0x5	Continue or Resume
STATUS_CLEAR_RTB	0x6	Clear the retransmit queue
STATUS_AUDIT	0x7	Audit state of link
STATUS_CONG_CLEAR	0x8	Congestion cleared
STATUS_CONG_ACCEPT	0x9	Congestion accept
STATUS_CONG_DISCARD	0xa	Congestion discard

3.3.1.6 State Confirm

The State Confirm message will be sent by the SGP in response to a State Request from the MGC. The State Confirm reflects that state value received in the State Request message.

The State Confirm message contains the following parameter:

State (mandatory)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x302)           |           Length = 8          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             State                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The valid values for State are shown in the following table. The value of the State field SHOULD reflect the value received in the State Request message.

Define	Value	Description
STATUS_LPO_SET	0x0	Request local processor outage
STATUS_LPO_CLEAR	0x1	Request local processor outage recovered
STATUS_EMER_SET	0x2	Request emergency alignment
STATUS_EMER_CLEAR	0x3	Request normal alignment (cancel emergency)
STATUS_FLUSH_BUFFERS	0x4	Flush or clear receive, transmit and retransmit queues
STATUS_CONTINUE	0x5	Continue or Resume
STATUS_CLEAR_RTB	0x6	Clear the retransmit queue
STATUS_AUDIT	0x7	Audit state of link
STATUS_CONG_CLEAR	0x8	Congestion cleared
STATUS_CONG_ACCEPT	0x9	Congestion accept
STATUS_CONG_DISCARD	0xa	Congestion discard

3.3.1.7 State Indication

The MTP2 State Indication message can be sent from a SGP to an ASP to indicate a condition on a SS7 link.

The State Indication message contains the following parameter:

Event (mandatory)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x303)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Event                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The valid values for Event are shown in the following table.

Define	Value	Description
EVENT_RPO_ENTER	0x1	Remote entered processor outage
EVENT_RPO_EXIT	0x2	Remote exited processor outage
EVENT_LPO_ENTER	0x3	Link entered processor outage
EVENT_LPO_EXIT	0x4	Link exited processor outage

3.3.1.8 Congestion Indication

The Congestion Indication message can be sent from a Signalling Gateway Process to an ASP to indicate the congestion status and discard status of a SS7 link. When the MSU buffer fill increases above an Onset threshold or decreases below an Abatement threshold or crosses a Discard threshold in either direction, the SGP SHALL send a congestion indication message when it supports SS7 MTP2 variants that support multiple congestion levels.

The SGP SHALL send the message only when there is actually a change in either the discard level or the congestion level to report, meaning it is different from the previously sent message. In addition, the SGP SHALL use an implementation dependent algorithm to limit the frequency of congestion indication messages.

An implementation may optionally send Congestion Indication messages on a "high priority" stream in order to potentially reduce delay.

The Congestion Indication message contains the following parameters:

Congestion Status (mandatory)
Discard Status (optional)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x304)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Congestion Status                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x305)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Discard Status                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The valid values for Congestion Status and Discard Status are shown in the following table.

Define	Value	Description
LEVEL_NONE	0x0	No congestion
LEVEL_1	0x1	Congestion Level 1
LEVEL_2	0x2	Congestion Level 2
LEVEL_3	0x3	Congestion Level 3

For SS7 networks that do not support multiple levels of congestion, only the LEVEL_NONE and LEVEL_3 values will be used. For SS7 networks that support multiple levels of congestion, it is possible for all values to be used. Refer to [2], [3] and [12] for more details on the Congestion and Discard Status of SS7 signalling links.

3.3.1.9 Retrieval Request

The MTP2 Retrieval Request message is used during the MTP Level 3 changeover procedure to request the BSN, to retrieve PDUs from the transmit and retransmit queues or to flush PDUs from the retransmit queue. Examples of the use of Retrieval Request for SS7 Link Changeover are provided in Section 5.3.6.

The Retrieval Request message contains the following parameters:

Action (mandatory)
Sequence Number (optional)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x306)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Action                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x307)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Sequence Number                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The valid values for Action are shown in the following table.

Define	Value	Description
ACTION_RTRV_BSN	0x1	Retrieve the backward sequence number
ACTION_RTRV_MSGS	0x2	Retrieve the PDUs from the transmit and retransmit queues

In the Retrieval Request message, the Sequence Number field SHOULD NOT be present if the Action field is ACTION_RTRV_BSN. The Sequence Number field contains the Forward Sequence Number (FSN) of the far end if the Action is ACTION_RTRV_MSGS.

3.3.1.10 Retrieval Confirm

The MTP2 Retrieval Confirm message is sent by the Signalling Gateway in response to a Retrieval Request message. Examples of the use of the Retrieval Confirm for SS7 Link Changeover are provided in Section 5.3.6.

The Retrieval Confirm message contains the following parameters:

Action (mandatory)
Result (mandatory)
Sequence Number (optional)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x306)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Action                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x308)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Result                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x307)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The valid values for Action are the same as in Retrieval Request.

The values for Result are shown below:

Define	Value	Description
RESULT_SUCCESS	0x0	Action successful
RESULT_FAILURE	0x1	Action failed

When the Signalling Gateway Process sends a Retrieval Confirm to a Retrieval Request, it echos the Action field. If the Action was ACTION_RTRV_BSN and the SGP successfully retrieved the BSN, the SGP will put the Backward Sequence Number (BSN) in the Sequence Number field and will indicate a success in the Result field. If the BSN could not be retrieved, the Sequence Number field will not be included and the Result field will indicate failure.

For a Retrieval Confirm with Action of ACTION_RTRV_MSGS, the value of the Result field will indicate success or failure. A failure means that the buffers could not be retrieved. The Sequence Number field is not used with ACTION_RTRV_MSGS.

3.3.1.11 Retrieval Indication

The Retrieval Indication message is sent by the Signalling Gateway with a PDU from the transmit or retransmit queue. The Retrieval Indication message does not contain the Action or Sequence Number fields, just a MTP3 Protocol Data Unit (PDU) from the transmit or retransmit queue. Examples of the use of the Retrieval Indication for SS7 Link Changeover are provided in Section 5.3.6.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag (0x300)          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                        Protocol Data                          /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

For TTC Data messages, the following parameter will be used to indicate a TTC PDU which starts at LI.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag (0x301)          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                      TTC Protocol Data                        /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The M2UA implementation MAY consider the use of the bundling feature of SCTP for Retrieval Indication messages.

3.3.1.12 Retrieval Complete Indication

The MTP2 Retrieval Complete Indication message is exactly the same as the MTP2 Retrieval Indication message except that it also indicates that retrieval is complete. In addition, it MAY contain a PDU (which MUST be the last PDU) from the transmit or retransmit queue.

3.3.2 Application Server Process Maintenance (ASPM) Messages

The ASPM messages will only use the common message header.

3.3.2.1 ASP Up (ASPUP)

The ASP Up (ASPUP) message is used to indicate to a remote M2UA peer that the Adaptation layer is ready to receive traffic or maintenance messages.

The ASPUP message contains the following parameters

ASP Identifier (optional)
Info String (optional)

Note: The ASP Identifier MUST be used where the SGP cannot identify the ASP by pre-configured address/port number information (e.g., where an ASP is resident on a Host using dynamic address/port number assignment).

The format for ASPUP Message parameters is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag (0x11)           |             Length = 8        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         ASP Identifier*                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag (0x4)            |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          INFO String*                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The optional ASP Identifier parameter would contain a unique value that is locally significant among the ASPs that support an AS. The SGP should save the ASP Identifier to be used, if necessary, with the Notify message (see Section 3.3.3.2).

The optional INFO String parameter can carry any meaningful UTF-8 [6] character string along with the message. Length of the INFO String parameter is from 0 to 255 octets. No procedures are presently identified for its use but the INFO String MAY be used for debugging purposes.

3.3.2.2 ASP Up Ack

The ASP Up Ack message is used to acknowledge an ASP Up message received from a remote M2UA peer.

The ASPUP Ack message contains the following parameters:

INFO String (optional)

The format for ASPUP Ack Message parameters is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag (0x4)            |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          INFO String*                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format and description of the optional Info String parameter is the same as for the ASP UP message (See Section 3.3.2.1).

3.3.2.3 ASP Down (ASPDN)

The ASP Down (ASPDN) message is used to indicate to a remote M2UA peer that the adaptation layer is not ready to receive traffic or maintenance messages.

The ASPDN message contains the following parameters

INFO String (optional)

The format for the ASPDN message parameters is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag (0x4)            |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          INFO String*                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format and description of the optional Info String parameter is the same as for the ASP Up message (See Section 3.3.2.1).

3.3.2.4 ASP Down Ack

The ASP Down Ack message is used to acknowledge an ASP Down message received from a remote M2UA peer.

The ASP Down Ack message contains the following parameters:

    INFO String (optional)

The format for the ASPDN Ack message parameters is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0x4)           |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          INFO String*                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format and description of the optional Info String parameter is the same as for the ASP UP message (See Section 3.3.2.1).

3.3.2.5 Heartbeat (BEAT)

The Heartbeat message is optionally used to ensure that the M2UA peers are still available to each other.

The BEAT message contains the following parameter:

    Heartbeat Data   Optional

The format for the BEAT message is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Tag = 0x0009       |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                       Heartbeat Data                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The sending node defines the Heartbeat Data field contents. It may include a Heartbeat Sequence Number and/or time stamp, or other implementation specific details.

The receiver of a Heartbeat message does not process this field as it is only of significance to the sender. The receiver echoes the content of the Heartbeat Data in a BEAT ACK message.

3.3.2.6 Heartbeat Ack (BEAT ACK)

The Heartbeat ACK message is sent in response to a BEAT message. A peer MUST send a BEAT ACK in response to a BEAT message. It includes all the parameters of the received Heartbeat message, without any change.

The BEAT ACK message contains the following parameter:

```
    Heartbeat Data           Optional
```

The format for the BEAT ACK message is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Tag = 0x0009       |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                       Heartbeat Data                          /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The sending node defines the Heartbeat Data field contents. It may include a Heartbeat Sequence Number and/or time stamp, or other implementation specific details.

The receiver of a Heartbeat message does not process this field as it is only of significance to the sender. The receiver echoes the content of the Heartbeat Data in a BEAT ACK message.

3.3.2.7 ASP Active (ASPAC)

The ASPAC message is sent by an ASP to indicate to an SGP that it is Active and ready to be used.

The ASPAC message contains the following parameters:

```
   Traffic Mode Type (optional)
   Interface Identifier (optional)
      - Combination of integer and integer ranges, OR
      - string (text formatted)
   INFO String (optional)
```

The format for the ASPAC message using integer formatted Interface Identifiers is as follows:

```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |           Tag (0xb)           |          Length = 8           |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                      Traffic Mode Type                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |     Tag (0x1=integer)         |            Length             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    /                                                               \
    \                      Interface Identifiers*                   /
    /                                                               \
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |    Tag (0x8=integer range)    |            Length             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Interface Identifier Start1*                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Interface Identifier Stop1*                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Interface Identifier Start2*                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Interface Identifier Stop2*                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      .                                                             .
      .                                                             .
      .                                                             .
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                  Interface Identifier StartN*                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Interface Identifier StopN*                 |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    /                                                               \
    \              Additional Interface Identifiers                 /
    /                    of Tag Type 0x1 or 0x8                     \
    \                                                               /
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Tag (0x4)             |             Length            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    /                                                               \
    \                          INFO String*                         /
    /                                                               \
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format for the ASPAC message using text formatted (string) Interface Identifiers is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0xb)           |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Traffic Mode Type                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Tag (0x3=string)        |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                      Interface Identifier*                    /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\              Additional Interface Identifiers                 /
/                       of Tag Type 0x3                         \
\                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x4)             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                           INFO String*                        /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Traffic Mode Type parameter identifies the traffic mode of operation of the ASP within an AS. The valid values for Type are shown in the following table:

Value	Description
0x1	Override
0x2	Load-share
0x3	Broadcast

Within a particular AS, only one Traffic Mode Type can be used. The Override value indicates that the ASP is operating in Override mode, where the ASP takes over all traffic in an Application Server (i.e., primary/backup operation), over-riding any currently active ASPs in the AS. In Load-share mode, the ASP will share in the traffic distribution with any other currently active ASPs. In Broadcast mode, all of the Active ASPs receive all message traffic in the Application Server.

The optional Interface Identifiers parameter contains a list of Interface Identifier integers (Type 0x1 or Type 0x8) or text strings (Type 0x3)indexing the Application Server traffic that the sending ASP is configured/registered to receive. If integer formatted Interface Identifiers are being used, the ASP can also send ranges of Interface Identifiers (Type 0x8). Interface Identifier types Integer (0x1) and Integer Range (0x8) are allowed in the same message. Text formatted Interface Identifiers (0x3) cannot be used with either Integer (0x1) or Integer Range (0x8) types.

If no Interface Identifiers are included, the message is for all provisioned Interface Identifiers within the AS(s) in which the ASP is provisioned. If only a subset of Interface Identifiers for an AS are included, the ASP is noted as Active for all the Interface Identifiers provisioned for that AS.

Note: If the optional Interface Identifier parameter is present, the integer formatted Interface Identifier MUST be supported, while the text formatted Interface Identifier MAY be supported.

An SGP that receives an ASPAC with an incorrect or unsupported Traffic Mode Type for a particular Interface Identifier will respond with an Error Message (Cause: Unsupported Traffic Handling Mode).

The format and description of the optional Info String parameter is the same as for the ASP UP message (See Section 3.3.2.1).

3.3.2.8 ASP Active Ack

The ASP Active (ASPAC) Ack message is used to acknowledge an ASP Active message received from a remote M2UA peer.

The ASPAC Ack message contains the following parameters:

```
Traffic Mode Type (optional)
Interface Identifier (optional)
   - Combination of integer and integer ranges, OR
   - string (text formatted)
INFO String (optional)
```

The format for the ASPAC Ack message with Integer-formatted Interface Identifiers is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0xb)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Traffic Mode Type                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Tag (0x1=integer)         |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                     Interface Identifiers*                    /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Tag (0x8=integer range)    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier Start1*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier Stop1*                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier Start2*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier Stop2*                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                                                             .
 .                                                             .
 .                                                             .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier StartN*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier StopN*                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\              Additional Interface Identifiers                 /
/                    of Tag Type 0x1 or 0x8                     \
\                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x4)             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          INFO String*                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format for the ASP Active Ack message using text formatted (string) Interface Identifiers is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0xb)           |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Traffic Mode Type                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Tag (0x3=string)        |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                      Interface Identifier*                    /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\               Additional Interface Identifiers                /
/                       of Tag Type 0x3                         \
\                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x4)             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          INFO String*                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format and description of the optional Info String parameter is the same as for the ASP Up message (See Section 3.3.2.1).

The format of the optional Interface Identifier parameter is the same as for the ASP Active message (See Section 3.3.2.7).

The format and description of the optional Info String parameter is the same as for the ASP Up message (See Section 3.3.2.1).

3.3.2.9 ASP Inactive (ASPIA)

The ASP Inactive (ASPIA) message is sent by an ASP to indicate to an SGP that it is no longer an active ASP to be used from within a list of ASPs. The SGP will respond with an ASPIA Ack message and either discard incoming messages or buffer for a timed period and then discard.

The ASPIA message contains the following parameters:

Interface Identifiers (optional)
- Combination of integer and integer ranges, OR
- string (text formatted)

INFO String (optional)

The format for the ASP Inactive message parameters using Integer formatted Interface Identifiers is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Tag (0x1=integer)         |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                      Interface Identifiers*                   /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Tag (0x8=integer range)    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier Start1*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Interface Identifier Stop1*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier Start2*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Interface Identifier Stop2*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                                                             .
 .                                                             .
 .                                                             .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier StartN*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Interface Identifier StopN*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\              Additional Interface Identifiers                 /
/                    of Tag Type 0x1 or 0x8                     \
\                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Tag (0x4)            |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          INFO String*                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format for the ASP Inactive message using text formatted (string) Interface Identifiers is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Tag (0x3=string)        |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                      Interface Identifier*                    /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\               Additional Interface Identifiers                /
/                       of Tag Type 0x3                         \
\                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x4)             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                           INFO String*                        /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the optional Interface Identifier parameter is the same as for the ASP Active message (See Section 3.3.2.7).

The format and description of the optional Info String parameter is the same as for the ASP Up message (See Section 3.3.2.1).

The optional Interface Identifiers parameter contains a list of Interface Identifier integers indexing the Application Server traffic that the sending ASP is configured/registered to receive, but does not want to receive at this time.

3.3.2.10 ASP Inactive Ack

The ASP Inactive (ASPIA) Ack message is used to acknowledge an ASP Inactive message received from a remote M2UA peer.

The ASPIA Ack message contains the following parameters:

Interface Identifiers (optional)
- Combination of integer and integer ranges, OR
- string (text formatted)

INFO String (optional)

The format for the ASPIA Ack message is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Tag (0x1=integer)         |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                      Interface Identifiers*                   /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Tag (0x8=integer range)    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier Start1*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Interface Identifier Stop1*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier Start2*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Interface Identifier Stop2*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 .                                                             .
 .                                                             .
 .                                                             .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Interface Identifier StartN*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Interface Identifier StopN*                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\               Additional Interface Identifiers                /
/                     of Tag Type 0x1 or 0x8                    \
\                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x4)             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                           INFO String*                        /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format for the ASP Inactive Ack message using text formatted (string) Interface Identifiers is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Tag (0x3=string)        |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                     Interface Identifier*                     /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\              Additional Interface Identifiers                 /
/                       of Tag Type 0x3                         \
\                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x4)             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          INFO String*                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the optional Interface Identifier parameter is the same as for the ASP Active message (See Section 3.3.2.7).

The format and description of the optional Info String parameter is the same as for the ASP Up message (See Section 3.3.2.1).

3.3.3 Layer Management (MGMT) Messages

3.3.3.1 Error (ERR)

The Error (ERR) message is used to notify a peer of an error event associated with an incoming message. For example, the message type might be unexpected given the current state, or a parameter value might be invalid.

An Error message MUST not be generated in response to other Error messages.

The ERR message contains the following parameters:

- Error Code (mandatory)
- Interface Identifier (optional)
- Diagnostic Information (optional)

The format for the ERR message is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0xc)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Error Code                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Tag (0x1, 0x3, or 0x8)    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                     Interface Identifier(s)*                  /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0x7)           |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                     Diagnostic Information*                   /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Error Code parameter indicates the reason for the Error Message. The Error parameter value can be one of the following values:

```
Invalid Version                          0x1
Invalid Interface Identifier             0x2
Unsupported Message Class                0x3
Unsupported Message Type                 0x4
Unsupported Traffic Handling Mode        0x5
Unexpected Message                       0x6
Protocol Error                           0x7
Unsupported Interface Identifier Type    0x8
Invalid Stream Identifier                0x9
Not Used in M2UA                         0xa
Not Used in M2UA                         0xb
Not Used in M2UA                         0xc
Refused - Management Blocking            0xd
ASP Identifier Required                  0xe
Invalid ASP Identifier                   0xf
ASP Active for Interface Identifier(s)   0x10
Invalid Parameter Value                  0x11
Parameter Field Error                    0x12
Unexpected Parameter                     0x13
Not Used in M2UA                         0x14
Not Used in M2UA                         0x15
Missing Parameter                        0x16
```

The "Invalid Version" error would be sent if a message was received with an invalid or unsupported version. The Error message would contain the supported version in the Common header. The Error message could optionally provide the supported version in the Diagnostic Information area.

The "Invalid Interface Identifier" error would be sent by a SGP if an ASP sends a message (i.e. an ASP Active message) with an invalid (not configured) Interface Identifier value. One of the optional Interface Identifier parameters (Integer-based, text-based or integer range) MUST be used with this error code to identify the invalid Interface Identifier(s) received.

The "Unsupported Traffic Handling Mode" error would be sent by a SGP if an ASP sends an ASP Active with an unsupported Traffic Handling Mode. An example would be a case in which the SGP did not support load-sharing. One of the optional Interface Identifier parameters (Integer-based, text-based or integer range) MAY be used with this error code to identify the Interface Identifier(s).

The "Unexpected Message" error would be sent by an ASP if it received a MAUP message from an SGP while it was in the Inactive state.

The "Protocol Error" error would be sent for any protocol anomaly (i.e. a bogus message).

The "Invalid Stream Identifier" error would be sent if a message was received on an unexpected SCTP stream (i.e. a MGMT message was received on a stream other than "0").

The "Unsupported Interface Identifier Type" error would be sent by a SGP if an ASP sends a Text formatted Interface Identifier and the SGP only supports Integer formatted Interface Identifiers. When the ASP receives this error, it will need to resend its message with an Integer formatted Interface Identifier.

The "Unsupported Message Class" error would be sent if a message with an unexpected or unsupported Message Class is received.

The "Refused - Management Blocking" error is sent when an ASP Up or ASP Active message is received and the request is refused for management reasons (e.g., management lock-out").

The "ASP Identifier Required" is sent by a SGP in response to an ASPUP message which does not contain an ASP Identifier parameter when the SGP requires one. The ASP SHOULD resend the ASPUP message with an ASP Identifier.

The "Invalid ASP Identifier" is sent by a SGP in response to an ASPUP message with an invalid (i.e. non-unique) ASP Identifier.

The "ASP Currently Active for Interface Identifier(s)" error is sent by a SGP when a Deregistration request is received from an ASP that is active for Interface Identifier(s) specified in the Deregistration request. One of the optional Interface Identifier parameters (Integer-based, text-based or integer range) MAY be used with this error code to identify the Interface Identifier(s).

The "Invalid Parameter Value " error is sent if a message is received with an invalid parameter value (e.g., a State Request with an an undefined State).

The "Parameter Field Error" would be sent if a message with a parameter has a wrong length field.

The "Unexpected Parameter" error would be sent if a message contains an invalid parameter.

The "Missing Parameter" error would be sent if a mandatory parameter was not included in a message.

The optional Diagnostic information can be any information germane to the error condition, to assist in the identification of the error condition. In the case of an Invalid Version Error Code the Diagnostic information includes the supported Version parameter. In the other cases, the Diagnostic information SHOULD be the first 40 bytes of the offending message.

3.3.3.2 Notify (NTFY)

The Notify message is used to provide an autonomous indication of M2UA events to an M2UA peer.

The NTFY message contains the following parameters:

- Status Type (mandatory)
- Status Information (mandatory)
- ASP Identifier (optional)
- Interface Identifiers (optional)
- INFO String (optional)

The format for the Notify message with Integer-formatted Interface Identifiers is as follows:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Tag (0xd)           |          Length = 8           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        Status Type            |      Status Information       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Tag (0x11)          |            Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         ASP Identifier*                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Tag (0x1=integer)         |            Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   /                                                               \
   \                      Interface Identifiers*                   /
   /                                                               \
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |    Tag (0x8=integer range)    |            Length             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Interface Identifier Start1*                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Interface Identifier Stop1*                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Interface Identifier Start2*                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Interface Identifier Stop2*                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    .                                                             .
    .                                                             .
    .                                                             .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Interface Identifier StartN*                 |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                  Interface Identifier StopN*                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   /                                                               \
   \              Additional Interface Identifiers                 /
   /                    of Tag Type 0x1 or 0x8                     \
   \                                                               /
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Tag (0x4)             |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   /                                                               \
   \                           INFO String*                        /
   /                                                               \
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format for the Notify message with Text-formatted Interface Identifiers is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0xd)           |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Status Type            |      Status Information       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Tag (0x11)          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        ASP Identifier*                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Tag (0x3=string)        |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                     Interface Identifier*                     /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\              Additional Interface Identifiers                 /
/                       of Tag Type 0x3                         \
\                                                               /
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag (0x4)             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
/                                                               \
\                          INFO String*                         /
/                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Status Type parameter identifies the type of the Notify message. The following are the valid Status Type values:

Value	Description
0x1	Application Server state change (AS_State_Change)
0x2	Other

The Status Information parameter contains more detailed information for the notification, based on the value of the Status Type. If the Status Type is AS_State_Change the following Status Information values are used:

Value	Description
1	reserved
2	Application Server Inactive (AS_Inactive)
3	Application Server Active (AS_Active)
4	Application Server Pending (AS_Pending)

These notifications are sent from an SGP to an ASP upon a change in status of a particular Application Server. The value reflects the new state of the Application Server. The Interface Identifiers of the AS MAY be placed in the message if desired.

If the Status Type is Other, then the following Status Information values are defined:

Value	Description
1	Insufficient ASP resources active in AS
2	Alternate ASP Active
3	ASP Failure

In the Insufficient ASP Resources case, the SGP is indicating to an ASP-INACTIVE ASP(s) in the AS that another ASP is required in order to handle the load of the AS (Load-sharing mode). For the Alternate ASP Active case, the formerly Active ASP is informed when an alternate ASP transitions to the ASP Active state in Override mode. The ASP Identifier (if available) of the Alternate ASP MUST be placed in the message. For the ASP Failure case, the SGP is indicating to ASP(s) in the AS that one of the ASPs has transitioned to ASP-DOWN. The ASP Identifier (if available) of the failed ASP MUST be placed in the message.

For each of the Status Information values in Status Type Other, the Interface Identifiers of the affected AS MAY be placed in the message if desired.

The format of the optional Interface Identifier parameter is the same as for the ASP Active message (See Section 3.3.2.7).

The format and description of the optional Info String parameter is the same as for the ASP Up message (See Section 3.3.2.1).

3.3.4 Interface Identifier Management (IIM) Messages

The Interface Identifier Management messages are optional. They are used to support the automatic allocation of Signalling Terminals or Signalling Data Links [2][3].

3.3.4.1 Registration Request (REG REQ)

The REG REQ message is sent by an ASP to indicate to a remote M2UA peer that it wishes to register one or more given Link Keys with the remote peer. Typically, an ASP would send this message to an SGP, and expect to receive a REG RSP in return with an associated Interface Identifier value.

The REG REQ message contains the following parameter:

Link Key (mandatory)

The format for the REG REQ message is as follows

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x0309          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                          Link Key 1                           /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                             ...                               /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x0309          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                          Link Key n                           /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Link Key: fixed length

The Link Key parameter is mandatory. The sender of this message expects that the receiver of this message will create a Link Key entry and assign a unique Interface Identifier value to it, if the Link Key entry does not yet exist.

The Link Key parameter may be present multiple times in the same message. This is used to allow the registration of multiple Link Keys in a single message.

The format of the Link Key parameter is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Local-LK-Identifier                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Signalling Data Terminal Identifier           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Signalling Data Link Identifier             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Local-LK-Identifier: 32-bit integer

The mandatory Local-LK-Identifier field is used to uniquely (between ASP and SGP) identify the registration request. The Identifier value is assigned by the ASP, and is used to correlate the response in a REG RSP message with the original registration request. The Identifier value MUST remain unique until the REG RSP is received.

The format of the Local-LK-Identifier field is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x030a          |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Local-LK-Identifier value                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Signalling Data Terminal Identifier

The Signalling Data Terminal Identifier parameter is mandatory. It identifies the Signalling Data Terminal associated with the SS7 link for which the ASP is registering. The format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x030b          |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved             |         SDT Identifier        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The SDT Identifier is a 32-bit unsigned value which may only be significant to 12 or 14 bits depending on the SS7 variant which is supported by the MTP Level 3 at the ASP. Insignificant SDT Identifier bits are coded 0.

Signalling Data Link Identifier

The Signalling Data Link Identifier parameter is mandatory. It identifies the Signalling Data Link Identifier associated with the SS7 link for which the ASP is registering. The format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x030c          |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved             |         SDL Identifier        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The SDL Identifier is a 32-bit unsigned value which may only be significant to 12 or 14 bits depending on the SS7 variant which is supported by the MTP Level 3 at the ASP. Insignificant SDLI bits are coded 0.

3.3.4.2 Registration Response (REG RSP)

The REG RSP message is used as a response to the REG REQ message from a remote M2UA peer. It contains indications of success/failure for registration requests and returns a unique Interface Identifier value for successful registration requests, to be used in subsequent M2UA Traffic Management protocol.

The REG RSP message contains the following parameter:

Registration Results (mandatory)

The format for the REG RSP message is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x030d          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                     Registration Result 1                     /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                              ...                              /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x030d          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                     Registration Result n                     /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Registration Results: fixed length

The Registration Results parameter contains one or more results, each containing the registration status for a single Link Key in the REG REQ message. The number of results in a single REG RSP message MAY match the number of Link Key parameters found in the corresponding REG REQ message. The format of each result is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Local-LK-Identifier                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Registration Status                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Interface Identifier                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Local-LK-Identifier: 32-bit integer

The Local-LK-Identifier contains the same value as found in the matching Link Key parameter found in the REG REQ message. The format of the Local-LK-Identifier is shown in Section 3.3.4.1.

Registration Status: 32-bit integer

The Registration Result Status field indicates the success or the reason for failure of a registration request.

Its values may be one of the following:

```
      0         Successfully Registered
      1         Error - Unknown
      2         Error - Invalid SDLI
      3         Error - Invalid SDTI
      4         Error - Invalid Link Key
      5         Error - Permission Denied
      6         Error - Overlapping (Non-unique) Link Key
      7         Error - Link Key not Provisioned
      8         Error - Insufficient Resources
```

The format of the Registration Status field is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x030e          |          Length = 8           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Registration Status                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Interface Identifier: 32-bit integer

The Interface Identifier field contains the Interface Identifier for the associated Link Key if the registration is successful. It is set to "0" if the registration was not successful. The format of integer-based and text-based Interface Identifier parameters are shown in Section 3.2.

3.3.4.3 De-Registration Request (DEREG REQ)

The DEREG REQ message is sent by an ASP to indicate to a remote M2UA peer that it wishes to de-register a given Interface Identifier. Typically, an ASP would send this message to an SGP, and expects to receive a DEREG RSP in return reflecting the Interface Identifier and containing a de-registration status.

The DEREG REQ message contains the following parameter:

Interface Identifier (mandatory)

The format for the DEREG REQ message is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Tag = 0x1 or 0x3          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                     Interface Identifier 1                    /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                              ...                              /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Tag = 0x1 or 0x3          |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                     Interface Identifier n                    /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Interface Identifier

The Interface Identifier parameter contains a Interface Identifier indexing the Application Server traffic that the sending ASP is currently registered to receive from the SGP but now wishes to de-register. The format of integer-based and text-based Interface Identifier parameters are shown in Section 3.2.

3.3.4.4 De-Registration Response (DEREG RSP)

The DEREG RSP message is used as a response to the DEREG REQ message from a remote M2UA peer.

The DEREG RSP message contains the following parameter:

De-Registration Results (mandatory)

The format for the DEREG RSP message is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x030f          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                   De-Registration Result 1                    /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                              ...                              /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x030f          |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
\                                                               \
/                   De-Registration Result n                    /
\                                                               \
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

De-Registration Results: fixed length

The De-Registration Results parameter contains one or more results, each containing the de-registration status for a single Interface Identifier in the DEREG REQ message. The number of results in a single DEREG RSP message MAY match the number of Interface Identifier parameters found in the corresponding DEREG REQ message. The format of each result is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Interface Identifier                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    De-Registration Status                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Interface Identifier: 32-bit integer

The Interface Identifier field contains the Interface Identifier value of the matching Link Key to de-register, as found in the DEREG REQ. The format of integer-based and text-based Interface Identifier parameters are shown in Section 3.2.

De-Registration Status: 32-bit integer

The De-Registration Result Status field indicates the success or the reason for failure of the de-registration.

Its values may be one of the following:

0	Successfully De-registered
1	Error - Unknown
2	Error - Invalid Interface Identifier
3	Error - Permission Denied
4	Error - Not Registered

The format of the De-Registration Status field is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Tag = 0x0310          |         Length = 8            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    De-Registration Status                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.0 Procedures

The M2UA layer needs to respond to various primitives it receives from other layers as well as messages it receives from the peer-to-peer messages. This section describes various procedures involved in response to these events.

4.1 Procedures to Support the M2UA-User Layer

These procedures achieve the M2UA layer "Transport of MTP Level 2 / MTP Level 3 boundary" service.

4.1.1 MTP Level 2 / MTP Level 3 Boundary Procedures

On receiving a primitive from the local upper layer, the M2UA layer will send the corresponding MAUP message (see Section 3) to its peer. The M2UA layer MUST fill in various fields of the common and specific headers correctly. In addition the message SHOULD be sent on the SCTP stream that corresponds to the SS7 link.

4.1.2 MAUP Message Procedures

On receiving MAUP messages from a peer M2UA layer, the M2UA layer on an SG or MGC needs to invoke the corresponding layer primitives to the local MTP Level 2 or MTP Level 3 layer.

4.2 Receipt of Primitives from the Layer Management

On receiving primitives from the local Layer Management, the M2UA layer will take the requested action and provide an appropriate response primitive to Layer Management.

An M-SCTP_ESTABLISH request primitive from Layer Management at an ASP will initiate the establishment of an SCTP association. The M2UA layer will attempt to establish an SCTP association with the remote M2UA peer by sending an SCTP-ASSOCIATE primitive to the local SCTP layer.

When an SCTP association has been successfully established, the SCTP will send an SCTP-COMMUNICATION_UP notification primitive to the local M2UA layer. At the SGP that initiated the request, the M2UA layer will send an M-SCTP_ESTABLISH confirm primitive to Layer Management when the association setup is complete. At the peer M2UA layer, an M-SCTP_ESTABLISH indication primitive is sent to Layer Management upon successful completion of an incoming SCTP association setup.

An M-SCTP_RELEASE request primitive from Layer Management initiates the shutdown of an SCTP association. The M2UA layer accomplishes a graceful shutdown of the SCTP association by sending an SCTP-SHUTDOWN primitive to the SCTP layer.

When the graceful shutdown of the SCTP association has been accomplished, the SCTP layer returns an SCTP-SHUTDOWN_COMPLETE notification primitive to the local M2UA layer. At the M2UA Layer that initiated the request, the M2UA layer will send an M-SCTP_RELEASE confirm primitive to Layer Management when the association shutdown is complete. At the peer M2UA Layer, an M-SCTP_RELEASE indication primitive is sent to Layer Management upon abort or successful shutdown of an SCTP association.

An M-SCTP_STATUS request primitive supports a Layer Management query of the local status of a particular SCTP association. The M2UA layer simply maps the M-SCTP_STATUS request primitive to an SCTP-STATUS primitive to the SCTP layer. When the SCTP responds, the M2UA layer maps the association status information to an M-SCTP_STATUS confirm primitive. No peer protocol is invoked.

Similar LM-to-M2UA-to-SCTP and/or SCTP-to-M2UA-to-LM primitive mappings can be described for the various other SCTP Upper Layer primitives in RFC 2960 [8] such as INITIALIZE, SET PRIMARY, CHANGE HEARTBEAT, REQUEST HEARTBEAT, GET SRTT REPORT, SET FAILURE THRESHOLD, SET PROTOCOL PARAMETERS, DESTROY SCTP INSTANCE, SEND FAILURE, AND NETWORK STATUS CHANGE. Alternatively, these SCTP Upper Layer

primitives (and Status as well) can be considered for modeling purposes as a Layer Management interaction directly with the SCTP Layer.

M-NOTIFY indication and M-ERROR indication primitives indicate to Layer Management the notification or error information contained in a received M2UA Notify or Error message respectively. These indications can also be generated based on local M2UA events.

An M-ASP_STATUS request primitive supports a Layer Management query of the status of a particular local or remote ASP. The M2UA layer responds with the status in an M-ASP_STATUS confirm primitive. No M2UA peer protocol is invoked.

An M-AS_STATUS request supports a Layer Management query of the status of a particular AS. The M2UA responds with an M-AS_STATUS confirm primitive. No M2UA peer protocol is invoked.

M-ASP_UP request, M-ASP_DOWN request, M-ASP_ACTIVE request and M-ASP_INACTIVE request primitives allow Layer Management at an ASP to initiate state changes. Upon successful completion, a corresponding confirm primitive is provided by the M2UA layer to Layer Management. If an invocation is unsuccessful, an Error indication primitive is provided in the primitive. These requests result in outgoing ASP Up, ASP Down, ASP Active and ASP Inactive messages to the remote M2UA peer at an SGP.

4.2.1 Receipt of M2UA Peer Management Messages

Upon successful state changes resulting from reception of ASP Up, ASP Down, ASP Active and ASP Inactive messages from a peer M2UA, the M2UA layer SHOULD invoke corresponding M-ASP_UP, M-ASP_DOWN, M-ASP_ACTIVE and M-ASP_INACTIVE, M-AS_ACTIVE, M-AS_INACTIVE, and M-AS_DOWN indication primitives to the local Layer Management.

M-NOTIFY indication and M-ERROR indication primitives indicate to Layer Management the notification or error information contained in a received M2UA Notify or Error message. These indications can also be generated based on local M2UA events.

All MGMT messages, except BEAT and BEAT Ack, SHOULD be sent with sequenced delivery to ensure ordering. All MGMT messages, with the exception of ASPTM, BEAT and BEAT Ack messages, SHOULD be sent on SCTP stream '0'. All ASPTM messages SHOULD be sent on the stream which normally carries the data traffic to which the message applies. BEAT and BEAT Ack messages MAY be sent using out-of-order delivery, and MAY be sent on any stream.

4.3 AS and ASP State Maintenance

The M2UA layer on the SGP maintains the state of each remote ASP, in each Application Server that the ASP is configured to receive traffic, as input to the M2UA message distribution function.

4.3.1 ASP States

The state of each remote ASP, in each AS that it is configured to operate, is maintained in the M2UA layer in the SGP. The state of a particular ASP in a particular AS changes due to events. The events include:

* Reception of messages from the peer M2UA layer at the ASP;
* Reception of some messages from the peer M2UA layer at other ASPs in the AS (e.g., ASP Active message indicating "Override");
* Reception of indications from the SCTP layer; or
* Local Management intervention.

The ASP state transition diagram is shown in Figure 5. The possible states of an ASP are:

ASP-DOWN: The remote M2UA peer at the ASP is unavailable and/or the related SCTP association is down. Initially all ASPs will be in this state. An ASP in this state SHOULD NOT be sent any M2UA messages, with the exception of Heartbeat, ASP Down Ack and Error messages.

ASP-INACTIVE: The remote M2UA peer at the ASP is available (and the related SCTP association is up) but application traffic is stopped. In this state the ASP MAY be sent any non-MAUP M2UA messages.

ASP-ACTIVE: The remote M2UA peer at the ASP is available and application traffic is active (for a particular Interface Identifier or set of Interface Identifiers).

Figure 5: ASP State Transition Diagram

```
                                      +--------------+
                                      |  ASP-ACTIVE  |
               +----------------------|              |
               |      Other   +-------|              |
               |   ASP in AS  |       +--------------+
               |   Overrides  |           ^     |
               |              |    ASP    |     | ASP
               |              |    Active |     | Inactive
               |              |           |     v
               |              |       +--------------+
               |              |       |              |
               |              +------>| ASP-INACTIVE |
               |                      +--------------+
               |                          ^     |
     ASP Down/ |                     ASP  |     | ASP Down /
     SCTP CDI/ |                     Up   |     | SCTP CDI/
     SCTP RI   |                          |     v SCTP RI
               |                      +--------------+
               |                      |              |
               +--------------------->|   ASP-DOWN   |
                                      |              |
                                      +--------------+
```

SCTP CDI: The SCTP CDI denotes the local SCTP layer's Communication Down Indication to the Upper Layer Protocol (M2UA) on an SGP. The local SCTP layer will send this indication when it detects the loss of connectivity to the ASP's peer SCTP layer. SCTP CDI is understood as either a SHUTDOWN_COMPLETE notification or COMMUNICATION_LOST notification from the SCTP layer.

SCTP RI: The local SCTP layer's Restart indication to the upper layer protocol (M2UA) on an SG. The local SCTP will send this indication when it detects a restart from the ASP's peer SCTP layer.

4.3.2 AS States

The state of the AS is maintained in the M2UA layer on the SGP. The state of an AS changes due to events. These events include:

* ASP state transitions
* Recovery timer triggers

The possible states of an AS are:

AS-DOWN: The Application Server is unavailable. This state implies that all related ASPs are in the ASP-DOWN state for this AS. Initially the AS will be in this state. An Application Server MUST be in the AS-DOWN state before it can be removed from a configuration.

AS-INACTIVE: The Application Server is available but no application traffic is active (i.e., one or more related ASPs are in the ASP-INACTIVE state, but none in the ASP-ACTIVE state). The recovery timer T(r) is not running or has expired.

AS-ACTIVE: The Application Server is available and application traffic is active. This state implies that at least one ASP is in the ASP-ACTIVE state.

AS-PENDING: An active ASP has transitioned to ASP-INACTIVE or ASP-DOWN and it was the last remaining active ASP in the AS. A recovery timer T(r) SHOULD be started and all incoming signalling messages SHOULD be queued by the SGP. If an ASP becomes ASP-ACTIVE before T(r) expires, the AS is moved to the AS-ACTIVE state and all the queued messages will be sent to the ASP.

If T(r) expires before an ASP becomes ASP-ACTIVE, the SGP stops queuing messages and discards all previously queued messages. The AS will move to the AS-INACTIVE state if at least one ASP is in the ASP-INACTIVE state, otherwise it will move to the AS-DOWN state.

Figure 6 shows an example AS state machine for the case where the AS/ASP data is pre-configured. For other cases where the AS/ASP configuration data is created dynamically, there would be differences in the state machine, especially at the creation of the AS.

For example, where the AS/ASP configuration data is not created until Registration of the first ASP, the AS-INACTIVE state is entered directly upon the first successful REG REQ from an ASP. Another example is where the AS/ASP configuration data is not created until the first ASP successfully enters the ASP-ACTIVE state. In this case the AS-ACTIVE state is entered directly.

```
                    Figure 6: AS State Transition Diagram

        +----------+   one ASP trans to ACTIVE   +-------------+
        |    AS-   |---------------------------->|     AS-     |
        | INACTIVE |                             |   ACTIVE    |
        |          |<---                         |             |
        +----------+    \                        +-------------+
           ^   |         \ Tr Expiry,                ^    |
           |   |          \ at least one             |    |
           |   |           \ ASP in ASP-INACTIVE     |    |
           |   |            \                        |    |
           |   |             \                       |    |
           |   |              \                      |    |
   one ASP |   | all ASP       \            one ASP  |    | Last ACTIVE
   trans   |   | trans to       \           trans to |    | ASP trans to
   to      |   | ASP-DOWN        -------\   ASP-     |    | ASP-INACTIVE
   ASP-    |   |                         \  ACTIVE   |    | or ASP-DOWN
   INACTIVE|   |                          \          |    | (start Tr)
           |   |                           \         |    |
           |   |                            \        |    |
           |   v                             \       |    v
        +----------+                          \  +-------------+
        |          |                           --|             |
        | AS-DOWN  |                             | AS-PENDING  |
        |          |                             |  (queuing)  |
        |          |<----------------------------|             |
        +----------+    Tr Expiry and no ASP     +-------------+
                        in ASP-INACTIVE state

      Tr = Recovery Timer
```

4.3.3 M2UA Management Procedures for Primitives

Before the establishment of an SCTP association the ASP state at both the SGP and ASP is assumed to be in the state ASP-DOWN.

Once the SCTP association is established (see Section 4.2.1) and assuming that the local M2UA-User is ready, the local M2UA ASP Maintenance (ASPM) function will initiate the relevant procedures, using the ASP Up/ASP Down/ASP Active/ASP Inactive messages to convey the ASP state to the SGP (see Section 4.3.4).

If the M2UA layer subsequently receives an SCTP-COMMUNICATION_DOWN or SCTP-RESTART indication primitive from the underlying SCTP layer, it will inform the Layer Management by invoking the M-SCTP_STATUS indication primitive. The state of the ASP will be moved to ASP-DOWN.

In the case of SCTP-COMMUNICATION_DOWN, the SCTP client MAY try to re-establish the SCTP association. This MAY be done by the M2UA layer automatically, or Layer Management MAY re-establish using the M-SCTP_ESTABLISH request primitive.

In the case of an SCTP-RESTART indication at an ASP, the ASP is now considered by its M2UA peer to be in the ASP-DOWN state. The ASP, if it is to recover, must begin any recovery with the ASP-Up procedure.

4.3.4 ASPM Procedures for Peer-to-Peer Messages

4.3.4.1 ASP Up Procedures

After an ASP has successfully established an SCTP association to an SGP, the SGP waits for the ASP to send an ASP Up message, indicating that the ASP M2UA peer is available. The ASP is always the initiator of the ASP Up message. This action MAY be initiated at the ASP by an M-ASP_UP request primitive from Layer Management or MAY be initiated automatically by an M2UA management function.

When an ASP Up message is received at an SGP and internally the remote ASP is in the ASP-DOWN state and not considered locked-out for local management reasons, the SGP marks the remote ASP in the state ASP-INACTIVE and informs Layer Management with an M-ASP_Up indication primitive. If the SGP is aware, via current configuration data, which Application Servers the ASP is configured to operate in, the SGP updates the ASP state to ASP-INACTIVE in each AS that it is a member.

Alternatively, the SGP may move the ASP into a pool of Inactive ASPs available for future configuration within Application Server(s), determined in a subsequent Registration Request or ASP Active procedure. If the ASP Up message contains an ASP Identifier, the SGP should save the ASP Identifier for that ASP. The SGP MUST send an ASP Up Ack message in response to a received ASP Up message even if the ASP is already marked as ASP-INACTIVE at the SGP.

If for any local reason (e.g., management lock-out) the SGP cannot respond with an ASP Up Ack message, the SGP responds to an ASP Up message with an Error message with Reason "Refused - Management Blocking".

At the ASP, the ASP Up Ack message received is not acknowledged. Layer Management is informed with an M-ASP_UP confirm primitive.

When the ASP sends an ASP Up message it starts timer T(ack). If the ASP does not receive a response to an ASP Up message within T(ack), the ASP MAY restart T(ack) and resend ASP Up messages until it

receives an ASP Up Ack message. T(ack) is provisionable, with a default of 2 seconds. Alternatively, retransmission of ASP Up messages MAY be put under control of Layer Management. In this method, expiry of T(ack) results in an M-ASP_UP confirm primitive carrying a negative indication.

The ASP MUST wait for the ASP Up Ack message before sending any other M2UA messages (e.g., ASP Active or REG REQ). If the SGP receives any other M2UA messages before an ASP Up message is received (other than ASP Down - see Section 4.3.4.2), the SGP MAY discard them.

If an ASP Up message is received and internally the remote ASP is in the ASP-ACTIVE state, an ASP Up Ack message is returned, as well as an Error message ("Unexpected Message), and the remote ASP state is changed to ASP-INACTIVE in all relevant Application Servers.

If an ASP Up message is received and internally the remote ASP is already in the ASP-INACTIVE state, an ASP Up Ack message is returned and no further action is taken.

4.3.4.1.1 M2UA Version Control

If an ASP Up message with an unsupported version is received, the receiving end responds with an Error message, indicating the version the receiving node supports and notifies Layer Management.

This is useful when protocol version upgrades are being performed in a network. A node upgraded to a newer version SHOULD support the older versions used on other nodes it is communicating with. Because ASPs initiate the ASP Up procedure it is assumed that the Error message would normally come from the SGP.

4.3.4.2 ASP Down Procedures

The ASP will send an ASP Down message to an SGP when the ASP wishes to be removed from service in all Application Servers that it is a member and no longer receive any MAUP or ASPTM messages. This action MAY be initiated at the ASP by an M-ASP_DOWN request primitive from Layer Management or MAY be initiated automatically by an M2UA management function.

Whether the ASP is permanently removed from any AS is a function of configuration management. In the case where the ASP previously used the Registration procedures (see Section 4.4) to register within Application Servers but has not unregistered from all of them prior to sending the ASP Down message, the SGP MUST consider the ASP as unregistered in all Application Servers that it is still a member.

The SGP marks the ASP as ASP-DOWN, informs Layer Management with an M-ASP_Down indication primitive, and returns an ASP Down Ack message to the ASP.

The SGP MUST send an ASP Down Ack message in response to a received ASP Down message from the ASP even if the ASP is already marked as ASP-DOWN at the SGP.

At the ASP, the ASP Down Ack message received is not acknowledged. Layer Management is informed with an M-ASP_DOWN confirm primitive. If the ASP receives an ASP Down Ack without having sent an ASP Down message, the ASP SHOULD now consider itself as in the ASP-DOWN state. If the ASP was previously in the ASP-ACTIVE or ASP_INACTIVE state, the ASP SHOULD then initiate procedures to return itself to its previous state.

When the ASP sends an ASP Down message it starts timer T(ack). If the ASP does not receive a response to an ASP Down message within T(ack), the ASP MAY restart T(ack) and resend ASP Down messages until it receives an ASP Down Ack message. T(ack) is provisionable, with a default of 2 seconds. Alternatively, retransmission of ASP Down messages MAY be put under control of Layer Management. In this method, expiry of T(ack) results in an M-ASP_DOWN confirm primitive carrying a negative indication.

4.3.4.3 ASP Active Procedures

Anytime after the ASP has received an ASP Up Ack message from the SGP, the ASP MAY send an ASP Active message to the SGP indicating that the ASP is ready to start processing traffic. This action MAY be initiated at the ASP by an M-ASP_ACTIVE request primitive from Layer Management or MAY be initiated automatically by a M2UA management function. In the case where an ASP wishes to process the traffic for more than one Application Server across a common SCTP association, the ASP Active message(s) SHOULD contain a list of one or more Interface Identifiers to indicate for which Application Servers the ASP Active message applies. It is not necessary for the ASP to include any Interface Identifiers of interest in a single ASP Active message, thus requesting to become active in all Interface Identifiers at the same time. Multiple ASP Active messages MAY be used to activate within the Application Servers independently, or in sets. In the case where an ASP Active message does not contain a Interface Identifier parameter, the receiver must know, via configuration data, of which Application Server(s) the ASP is a member.

For the Application Servers that the ASP can successfully activate, the SGP responds with one or more ASP Active Ack messages, including

the associated Interface Identifier(s) and reflecting any Traffic Mode Type value present in the related ASP Active message. The Interface Identifier parameter MUST be included in the ASP Active Ack message(s) if the received ASP Active message contained any Interface Identifiers. Depending on any Traffic Mode Type request in the ASP Active message or local configuration data if there is no request, the SGP moves the ASP to the correct ASP traffic state within the associated Application Server(s). Layer Management is informed with an M-ASP_Active indication. If the SGP receives any Data messages before an ASP Active message is received, the SGP MAY discard them. By sending an ASP Active Ack message, the SGP is now ready to receive and send traffic for the related Interface Identifier(s). The ASP SHOULD NOT send MAUP messages for the related Interface Identifier(s) before receiving an ASP Active Ack message, or it will risk message loss.

Multiple ASP Active Ack messages MAY be used in response to an ASP Active message containing multiple Interface Identifiers, allowing the SGP to independently acknowledge the ASP Active message for different (sets of) Interface Identifiers. The SGP MUST send an Error message ("Invalid Interface Identifier") for each Interface Identifier value that cannot be successfully activated.

In the case where an "out-of-the-blue" ASP Active message is received (i.e., the ASP has not registered with the SG or the SG has no static configuration data for the ASP), the message MAY be silently discarded.

The SGP MUST send an ASP Active Ack message in response to a received ASP Active message from the ASP, if the ASP is already marked in the ASP-ACTIVE state at the SGP.

At the ASP, the ASP Active Ack message received is not acknowledged. Layer Management is informed with an M-ASP_ACTIVE confirm primitive. It is possible for the ASP to receive Data message(s) before the ASP Active Ack message as the ASP Active Ack and Data messages from an SG may be sent on different SCTP streams. Message loss is possible as the ASP does not consider itself in the ASP-ACTIVE state until reception of the ASP Active Ack message.

When the ASP sends an ASP Active message it starts timer T(ack). If the ASP does not receive a response to an ASP Active message within T(ack), the ASP MAY restart T(ack) and resend ASP Active message(s) until it receives an ASP Active Ack message. T(ack) is provisionable, with a default of 2 seconds. Alternatively, retransmission of ASP Active messages MAY be put under the control of Layer Management. In this method, expiry of T(ack) results in an M-ASP_ACTIVE confirm primitive carrying a negative indication.

There are three modes of Application Server traffic handling in the SGP M2UA layer: Override, Load share and Broadcast. When included, the Traffic Mode Type parameter in the ASP Active message indicates the traffic handling mode to be used in a particular Application Server. If the SGP determines that the mode indicated in an ASP Active message is unsupported or incompatible with the mode currently configured for the AS, the SGP responds with an Error message ("Unsupported / Invalid Traffic Handling Mode"). If the traffic handling mode of the Application Server is not already known via configuration data, the traffic handling mode indicated in the first ASP Active message causing the transition of the Application Server state to AS-ACTIVE MAY be used to set the mode.

In the case of an Override mode AS, reception of an ASP Active message at an SGP causes the (re)direction of all traffic for the AS to the ASP that sent the ASP Active message. Any previously active ASP in the AS is now considered to be in the state ASP-INACTIVE and SHOULD no longer receive traffic from the SGP within the AS. The SGP then MUST send a Notify message ("Alternate ASP Active") to the previously active ASP in the AS, and SHOULD stop traffic to/from that ASP. The ASP receiving this Notify MUST consider itself now in the ASP-INACTIVE state, if it is not already aware of this via inter-ASP communication with the Overriding ASP.

In the case of a Load-share mode AS, reception of an ASP Active message at an SGP causes the direction of traffic to the ASP sending the ASP Active message, in addition to all the other ASPs that are currently active in the AS. The algorithm at the SGP for load-sharing traffic within an AS to all the active ASPs is implementation dependent. The algorithm could, for example be round-robin or based on information in the Data message (e.g., such as the SLS in the Routing Label).

An SGP, upon reception of an ASP Active message for the first ASP in a Load share AS, MAY choose not to direct traffic to a newly active ASP until it determines that there are sufficient resources to handle the expected load (e.g., until there are "n" ASPs in state ASP-ACTIVE in the AS).

All ASPs within a load-sharing mode AS must be able to process any Data message received for the AS, to accommodate any potential fail-over or balancing of the offered load.

In the case of a Broadcast mode AS, reception of an ASP Active message at an SGP causes the direction of traffic to the ASP sending the ASP Active message, in addition to all the other ASPs that are currently active in the AS. The algorithm at the SGP for

broadcasting traffic within an AS to all the active ASPs is a simple broadcast algorithm, where every message is sent to each of the active ASPs.

An SGP, upon reception of an ASP Active message for the first ASP in a Broadcast AS, MAY choose not to direct traffic to a newly active ASP until it determines that there are sufficient resources to handle the expected load (e.g., until there are "n" ASPs in state ASP-ACTIVE in the AS).

Whenever an ASP in a Broadcast mode AS becomes ASP-ACTIVE, the SGP MUST tag the first DATA message broadcast in each SCTP stream with a unique Correlation Id parameter. The purpose of this Correlation Id is to permit the newly active ASP to synchronize its processing of traffic in each ordered stream with the other ASPs in the broadcast group.

4.3.4.4 ASP Inactive Procedures

When an ASP wishes to withdraw from receiving traffic within an AS, the ASP sends an ASP Inactive message to the SGP. This action MAY be initiated at the ASP by an M-ASP_INACTIVE request primitive from Layer Management or MAY be initiated automatically by an M2UA management function. In the case where an ASP is processing the traffic for more than one Application Server across a common SCTP association, the ASP Inactive message contains one or more Interface Identifiers to indicate for which Application Servers the ASP Inactive message applies. In the case where an ASP Inactive message does not contain a Interface Identifier parameter, the receiver must know, via configuration data, of which Application Servers the ASP is a member and move the ASP to the ASP-INACTIVE state in all Application Servers. In the case of an Override mode AS, where another ASP has already taken over the traffic within the AS with an ASP Active ("Override") message, the ASP that sends the ASP Inactive message is already considered by the SGP to be in the state ASP-INACTIVE. An ASP Inactive Ack message is sent to the ASP, after ensuring that all traffic is stopped to the ASP.

In the case of a Load-share mode AS, the SGP moves the ASP to the ASP-INACTIVE state and the AS traffic is re-allocated across the remaining ASPs in the state ASP-ACTIVE, as per the load-sharing algorithm currently used within the AS. A Notify message ("Insufficient ASP resources active in AS") MAY be sent to all inactive ASPs, if required. An ASP Inactive Ack message is sent to the ASP after all traffic is halted and Layer Management is informed with an M-ASP_INACTIVE indication primitive.

In the case of a Broadcast mode AS, the SGP moves the ASP to the ASP-INACTIVE state and the AS traffic is broadcast only to the remaining ASPs in the state ASP-ACTIVE. A Notify message ("Insufficient ASP resources active in AS") MAY be sent to all inactive ASPs, if required. An ASP Inactive Ack message is sent to the ASP after all traffic is halted and Layer Management is informed with an M-ASP_INACTIVE indication primitive.

Multiple ASP Inactive Ack messages MAY be used in response to an ASP Inactive message containing multiple Interface Identifiers, allowing the SGP to independently acknowledge for different (sets of) Interface Identifiers. The SGP sends an Error message ("Invalid Interface Identifier") for each invalid or not configured Interface Identifier value in a received ASP Inactive message.

The SGP MUST send an ASP Inactive Ack message in response to a received ASP Inactive message from the ASP and the ASP is already marked as ASP-INACTIVE at the SGP.

At the ASP, the ASP Inactive Ack message received is not acknowledged. Layer Management is informed with an M-ASP_INACTIVE confirm primitive. If the ASP receives an ASP Inactive Ack without having sent an ASP Inactive message, the ASP SHOULD now consider itself as in the ASP-INACTIVE state. If the ASP was previously in the ASP-ACTIVE state, the ASP SHOULD then initiate procedures to return itself to its previous state.

When the ASP sends an ASP Inactive message it starts timer T(ack). If the ASP does not receive a response to an ASP Inactive message within T(ack), the ASP MAY restart T(ack) and resend ASP Inactive messages until it receives an ASP Inactive Ack message. T(ack) is provisionable, with a default of 2 seconds. Alternatively, retransmission of ASP Inactive messages MAY be put under the control of Layer Management. In this method, expiry of T(ack) results in a M-ASP_Inactive confirm primitive carrying a negative indication.

If no other ASPs in the Application Server are in the state ASP-ACTIVE, the SGP MUST send a Notify message ("AS-Pending") to all of the ASPs in the AS which are in the state ASP-INACTIVE. The SGP SHOULD start buffering the incoming messages for T(r)seconds, after which messages MAY be discarded. T(r) is configurable by the network operator. If the SGP receives an ASP Active message from an ASP in the AS before expiry of T(r), the buffered traffic is directed to that ASP and the timer is canceled. If T(r) expires, the AS is moved to the AS-INACTIVE state.

4.3.4.5 Notify Procedures

A Notify message reflecting a change in the AS state MUST be sent to all ASPs in the AS, except those in the ASP-DOWN state, with appropriate Status Information and any ASP Identifier of the failed ASP. At the ASP, Layer Management is informed with an M-NOTIFY indication primitive. The Notify message MUST be sent whether the AS state change was a result of an ASP failure or reception of an ASP State Management (ASPSM) / ASP Traffic Management (ASPTM) message. In the second case, the Notify message MUST be sent after any related acknowledgment messages (e.g., ASP Up Ack, ASP Down Ack, ASP Active Ack, or ASP Inactive Ack).

In the case where a Notify ("AS-PENDING") message is sent by an SGP that now has no ASPs active to service the traffic, or where a Notify ("Insufficient ASP resources active in AS") message MUST be sent in the Load share or Broadcast mode, the Notify message does not explicitly compel the ASP(s) receiving the message to become active. The ASPs remain in control of what (and when) traffic action is taken.

In the case where a Notify message does not contain a Interface Identifier parameter, the receiver must know, via configuration data, of which Application Servers the ASP is a member and take the appropriate action in each AS.

4.3.4.6 Heartbeat Procedures

The optional Heartbeat procedures MAY be used when operating over transport layers that do not have their own heartbeat mechanism for detecting loss of the transport association (i.e., other than SCTP).

Either M2UA peer may optionally send Heartbeat messages periodically, subject to a provisionable timer T(beat). Upon receiving a Heartbeat message, the M2UA peer MUST respond with a Heartbeat Ack message.

If no Heartbeat Ack message (or any other M2UA message) is received from the M2UA peer within 2*T(beat), the remote M2UA peer is considered unavailable. Transmission of Heartbeat messages is stopped and the signalling process SHOULD attempt to re-establish communication if it is configured as the client for the disconnected M2UA peer.

The Heartbeat message may optionally contain an opaque Heartbeat Data parameter that MUST be echoed back unchanged in the related Heartbeat Ack message. The sender, upon examining the contents of the returned Heartbeat Ack message, MAY choose to consider the remote M2UA peer as unavailable. The contents/format of the Heartbeat Data parameter is

implementation-dependent and only of local interest to the original sender. The contents may be used, for example, to support a Heartbeat sequence algorithm (to detect missing Heartbeats), and/or a time stamp mechanism (to evaluate delays).

Note: Heartbeat related events are not shown in Figure 5 "ASP state transition diagram".

4.4 Link Key Management Procedures

The Interface Identifier Management procedures are optional. They can be used to support automatic allocation of Signalling Terminals or Signalling Data Links [2][3].

4.4.1 Registration

An ASP MAY dynamically register with an SGP as an ASP within an Application Server for individual Interface Identifier(s) using the REG REQ message. A Link Key parameter in the REG REQ specifies the parameters associated with the Link Key.

The SGP examines the contents of the received Link Key parameters (SDLI and SDTI) and compares them with the currently provisioned Interface Identifiers. If the received Link Key matches an existing SGP Link Key entry, and the ASP is not currently included in the list of ASPs for the related Application Server, the SGP MAY authorize the ASP to be added to the AS. Or, if the Link Key does not currently exist and the received Link Key data is valid and unique, an SGP supporting dynamic configuration MAY authorize the creation of a new Interface Identifier and related Application Server and add the ASP to the new AS. In either case, the SGP returns a Registration Response message to the ASP, containing the same Local-LK-Identifier as provided in the initial request, a Registration Result "Successfully Registered" and the Interface Identifier. A unique method of Interface Identifier valid assignment at the SG/SGP is implementation dependent but MUST be guaranteed to be unique for each Application server or Link Key served by SGP.

If the SGP determines that the received Link Key data is invalid, or contains invalid parameter values, the SGP returns a Registration Response message to the ASP, containing a Registration Result "Error - Invalid Link Key", "Error - Invalid SDTI", "Error - Invalid SDLI" as appropriate.

If the SGP determines that the Link Key parameter overlaps with an existing Link Key entry, the SGP returns a Registration Response message to the ASP, with a Registration Status of "Error - Overlapping (Non-Unique) Link Key". An incoming signalling message received at an SGP cannot match against more than one Link Key.

If the SGP does not authorize the registration request, the SGP returns a REG RSP message to the ASP containing the Registration Result "Error - Permission Denied".

If an SGP determines that a received Link Key does not currently exist and the SGP does not support dynamic configuration, the SGP returns a Registration Response message to the ASP, containing a Registration Result "Error - Link Key not Provisioned".

If an SGP determines that a received Link Key does not currently exist and the SGP supports dynamic reconfiguration but does not have the capacity to add new Link Key and Application Server entries, the SGP returns a Registration Response message to the ASP, containing a Registration Result "Error - Insufficient Resources".

An ASP MAY register multiple Link Keys at once by including a number of Link Key parameters in a single REG REQ message. The SGP MAY respond to each registration request in a single REG RSP message, indicating the success or failure result for each Link Key in a separate Registration Result parameter. Alternatively, the SGP MAY respond with multiple REG RSP messages, each with one or more Registration Result parameters. The ASP uses the Local-LK-Identifier parameter to correlate the requests with the responses.

4.4.2 Deregistration

An ASP MAY dynamically de-register with an SGP as an ASP within an Application Server for individual Interface Identifier(s) using the DEREG REQ message. A Interface Identifier parameter in the DEREG REQ specifies which Interface Identifier to de-register.

The SGP examines the contents of the received Interface Identifier parameter and validates that the ASP is currently registered in the Application Server(s) related to the included Interface Identifier(s). If validated, the ASP is de-registered as an ASP in the related Application Server.

The deregistration procedure does not necessarily imply the deletion of Link Key and Application Server configuration data at the SGP. Other ASPs may continue to be associated with the Application Server,

in which case the Link Key data CANNOT be deleted. If a Deregistration results in no more ASPs in an Application Server, an SGP MAY delete the Link Key data.

The SGP acknowledges the de-registration required by returning a DEREG RSP to the requesting ASP. The result of the de-registration is found in the Deregistration Result parameter, indicating success or failure with cause.

An ASP MAY de-register multiple Interface Identifiers at once by including a number of Interface Identifiers in a single DEREG REQ message. The SGP MUST respond to each deregistration request in a single DEREG RSP message, indicating the success or failure result for each Interface Identifier in a separate Deregistration Result parameter.

5.0 Examples of MTP2 User Adaptation (M2UA) Procedures

5.1 Establishment of associations between SGP and MGC examples

5.1.1 Single ASP in an Application Server (1+0 sparing)

This scenario shows the example M2UA message flows for the establishment of traffic between an SGP and an ASP, where only one ASP is configured within an AS (no backup). It is assumed that the SCTP association is already set-up.

```
    SGP                          ASP1
     |
     |<---------ASP Up----------|
     |--------ASP Up Ack------->|
     |                          |
     |<-------ASP Active--------|
     |------ASP Active Ack----->|
     |                          |
     |------NTFY(AS-ACTIVE)---->|
```

5.1.2 Single ASP in an Application Server (1+0 sparing) with Dynamic Registration

This scenario is the same as the one shown in Section 5.1.1 except with a dynamic registration (automatic allocation) of an Interface Identifier(s).

```
                SGP                         ASP1
                 |
                 |<---------ASP Up----------|
                 |--------ASP Up Ack------->|
                 |                          |
                 |<--------REG REQ----------|
                 |------REG REQ RESP------->|
                 |                          |
                 |<-------ASP Active--------|
                 |------ASP Active Ack----->|
                 |                          |
                 |------NTFY(AS-ACTIVE)---->|
```

5.1.3 Two ASPs in Application Server (1+1 sparing)

This scenario shows the example M2UA message flows for the establishment of traffic between an SGP and two ASPs in the same Application Server, where ASP1 is configured to be active and ASP2 to be standby in the event of communication failure or the withdrawal from service of ASP1. ASP2 MAY act as a hot, warm, or cold standby depending on the extent to which ASP1 and ASP2 share call/transaction state or can communicate call state under failure/withdrawal events.

```
          SGP                        ASP1                        ASP2
           |                          |                           |
           |<--------ASP Up----------|                           |
           |-------ASP Up Ack------->|                           |
           |                          |                           |
           |<-----------------------------ASP Up----------------|
           |----------------------------ASP Up Ack------------->|
           |                          |                           |
           |                          |                           |
           |<-------ASP Active-------|                           |
           |-----ASP Active Ack----->|                           |
           |                          |                           |
           |                          |                           |
           |-----NTFY(AS-ACTIVE)---->|                           |
           |                          |                           |
           |------------------NTFY(AS-ACTIVE)------------------>|
```

5.2 ASP Traffic Fail-over Examples

5.2.1 (1+1 Sparing, withdrawal of ASP, backup Override)

Following on from the example in Section 5.1.2, and ASP withdraws from service:

```
         SGP                       ASP1                       ASP2
           |                         |                          |
           |<-----ASP Inactive-------|                          |
           |----ASP Inactive Ack---->|                          |
           |                         |                          |
           |----NTFY(AS-PENDING)---->|                          |
           |------------------NTFY(AS-PENDING)----------------->|
           |                         |                          |
           |<------------------------------ ASP Active----------|
           |-----------------------------ASP Active Ack-------->|
           |                         |                          |
           |-----NTFY(AS-ACTIVE)---->|                          |
           |------------------NTFY(AS-ACTIVE)------------------>|
           |                         |                          |
```

In this case, the SGP notifies ASP2 that the AS has moved to the AS-PENDING state. ASP2 sends ASP Active to bring the AS back to the AS-ACTIVE state. If ASP2 did not send the ASP Active message before T(r) expired, the SGP would send a NOTIFY (AS-DOWN).

Note: If the SGP detects loss of the M2UA peer (through a detection of SCTP failure), the initial SGP-ASP1 ASP Inactive message exchange would not occur.

```
          SGP                       ASP1                       ASP2
           |                         |                          |
     (detects SCTP failure)
           |------------------NTFY(AS-PENDING)----------------->|
           |                         |                          |
           |<------------------------------ ASP Active----------|
           |-----------------------------ASP Active Ack-------->|
           |                         |                          |
           |------------------NTFY(AS-ACTIVE)------------------>|
           |                         |                          |
```

5.2.2 (1+1 Sparing, backup Override)

Following on from the example in Section 5.1.2, and ASP2 wishes to override ASP1 and take over the traffic:

```
      SGP                       ASP1                       ASP2
       |                          |                          |
       |<------------------------------ASP Active----------|
       |----------------------------ASP Active Ack-------->|
       |----NTFY(Alt ASP-Act)--->|                          |
       |                          |                          |
```

In this case, the SGP notifies ASP1 that an alternative ASP has overridden it.

5.3 SGP to MGC, MTP Level 2 to MTP Level 3 Boundary Procedures

When the M2UA layer on the ASP has a MAUP message to send to the SGP, it will do the following:

- Determine the correct SGP

- Find the SCTP association to the chosen SGP

- Determine the correct stream in the SCTP association based on the SS7 link

- Fill in the MAUP message, fill in M2UA Message Header, fill in Common Header

- Send the MAUP message to the remote M2UA peer in the SGP, over the SCTP association

When the M2UA layer on the SGP has a MAUP message to send to the ASP, it will do the following:

- Determine the AS for the Interface Identifier

- Determine the Active ASP (SCTP association) within the AS

- Determine the correct stream in the SCTP association based on the SS7 link

- Fill in the MAUP message, fill in M2UA Message Header, fill in Common Header

- Send the MAUP message to the remote M2UA peer in the ASP, over the SCTP association

5.3.1 SS7 Link Alignment

The MGC can request that a SS7 link be brought into alignment using the normal or emergency procedure [2][3]. An example of the message flow to bring a SS7 link in-service using the normal alignment procedure is shown below.

```
       MTP2            M2UA                    M2UA              MTP3
        SGP             SGP                     ASP               ASP

        <----Start Req---|<---Establish Req----|<----Start Req------

        ---In Serv Ind-->|----Establish Cfm--->|----In Serv Ind---->
```

An example of the message flow to bring a SS7 link in-service using the emergency alignment procedure.

```
 MTP2            M2UA                                  M2UA           MTP3
  SGP             SGP                                   ASP            ASP

  <----Emer Req----|<--State Req (STATUS_EMER_SET)----|<----Emer Req---

  -----Emer Cfm--->|---State Cfm (STATUS_EMER_SET)--->|----Emer Cfm---->

  <---Start Req----|<-------Establish Req-------------|<---Start Req----

  ---In Serv Ind-->|--------Establish Cfm------------>|---In Serv Ind-->
```

5.3.2 SS7 Link Release

The MGC can request that a SS7 link be taken out-of-service. It uses the Release Request message as shown below.

```
     MTP2                  M2UA                  M2UA                MTP3
      SGP                   SGP                   ASP                 ASP

       <-----Stop Req-----|<---Release Req------|<-----Stop Req------

       --Out of Serv Ind->|----Release Cfm----->|--Out of Serv Ind-->
```

The SGP can autonomously indicate that a SS7 link has gone out-of-service as shown below.

```
       MTP2           M2UA                      M2UA              MTP3
        SGP            SGP                       ASP               ASP

         --Out of Serv->|----Release Ind----->|--Out of Serv-->
```

5.3.3 Set and Clear Local Processor Outage

The MGC can set a Local Processor Outage condition. It uses the State Request message as shown below.

```
  MTP2            M2UA                                  M2UA           MTP3
   SGP             SGP                                   ASP            ASP

   <----LPO Req----|<---State Req (STATUS_LPO_SET)----|<----LPO Req---

   -----LPO Cfm--->|----State Cfm (STATUS_LPO_SET)--->|----LPO Cfm---->
```

The MGC can clear a Local Processor Outage condition. It uses the State Request message as shown below.

```
  MTP2            M2UA                                  M2UA          MTP3
   SGP             SGP                                   ASP           ASP

   <---LPO Req---|<---State Req (STATUS_LPO_CLEAR)----|<----LPO Req---

   ----LPO Cfm-->|----State Cfm (STATUS_LPO_CLEAR)--->|----LPO Cfm---->
```

5.3.4 Notification of Remote Processor Outage

The SGP can indicate that Remote has entered or exited the Processor Outage condition for a SS7 link. It uses the State Indication message as shown below.

```
 MTP2            M2UA                                  M2UA           MTP3
  SGP             SGP                                   ASP            ASP

  ----RPO Ind---->|----State Ind (EVENT_RPO_ENTER)-->|-----RPO Ind---->

  -RPO Rcvr Ind-->|----State Ind (EVENT_RPO_EXIT)--->|--RPO Rcvr Ind-->
```

5.3.5 Notification of SS7 Link Congestion

The SGP can indicate that a SS7 link has become congested. It uses the Congestion Indication message as shown below.

```
 MTP2           M2UA                                 M2UA           MTP3
  SGP            SGP                                  ASP            ASP

  ----Cong Ind---->|--------Cong Ind (STATUS)------->|----Cong Ind---->

  -Cong Cease Ind->|--------Cong Ind (STATUS)------->|-Cong Cease Ind->
```

5.3.6 SS7 Link Changeover

An example of the message flow for an error free changeover is shown below. In this example, there were three messages in the retransmission queue that needed to be retrieved.

```
  MTP2          M2UA                             M2UA             MTP3
   SGP           SGP                              ASP              ASP

   <-Rtrv BSN Req-|<--Rtrv Req (ACTION_RTRV_BSN)--|<--Rtrv BSN Req---
                              (seq_num = 0)

   -Rtrv BSN Cfm->|---Rtrv Cfm (ACTION_RTRV_BSN)->|---Rtrv BSN Cfm-->
                              (seq_num = BSN)

   <-Rtrv Msg Req-|<-Rtrv Req (ACTION_RTRV_MSGS)--|<--Rtrv Msg Req---
                              (seq_num = FSN)

   -Rtrv Msg Cfm->|--Rtrv Cfm (ACTION_RTRV_MSGS)->|---Rtrv Msg Cfm-->
                              (seq_num = 0)

   -Rtrv Msg Ind->|---------Retrieval Ind ------->|---Rtrv Msg Ind-->
   -Rtrv Msg Ind->|---------Retrieval Ind ------->|---Rtrv Msg Ind-->
   -Rtrv Msg Ind->|---------Retrieval Ind ------->|---Rtrv Msg Ind-->

   -Rtrv Compl Ind->|----Retrieval Compl Ind ---->|-Rtrv Compl Ind-->
```

Note: The number of Retrieval Indication is dependent on the number of messages in the retransmit queue that have been requested. Only one Retrieval Complete Indication SHOULD be sent.

An example of a message flow with an error retrieving the BSN is shown below.

```
  MTP2          M2UA                               M2UA             MTP3
   SGP           SGP                                ASP              ASP

   <-Rtrv BSN Req-|<--Rtrv Req (ACTION_RTRV_BSN)--|<--Rtrv BSN Req---

   -BSN Not Rtrv->|---Rtrv Cfm (ACTION_RTRV_BSN)->|---BSN Not Rtrv-->
                               (seq_num = -1)
```

An example of a message flow with an error retrieving the messages is shown below.

```
   <-Rtrv BSN Req-|<--Rtrv Req (ACTION_RTRV_BSN)--|<--Rtrv BSN Req---

   -Rtrv BSN Cfm->|---Rtrv Cfm (ACTION_RTRV_BSN)->|---Rtrv BSN Cfm-->
                               (seq_num = BSN)

   <-Rtrv Msg Req-|<-Rtrv Req (ACTION_RTRV_MSGS)--|<--Rtrv Msg Req---
                               (seq_num = FSN)

   -Rtrv Msg Cfm->|--Rtrv Cfm (ACTION_RTRV_MSGS)->|---Rtrv Msg Cfm-->
                               (seq_num = -1)
```

An example of a message flow for a request to drop messages (clear retransmission buffers) is shown below.

```
  MTP2          M2UA                               M2UA             MTP3
   SGP           SGP                                ASP              ASP

 -Clr RTB Req----|<-StateReq (STATUS_CLEAR_RTB)--|<--Clr RTB Req-----

 -Clr RTB Req--->|-StateCfm (STATUS_CLEAR_RTB)-->|---Clr RTB Req---->
```

5.3.7 Flush and Continue

The following message flow shows a request to flush buffers.

```
  MTP2            M2UA                                 M2UA          MTP3
   SGP             SGP                                  ASP           ASP

   <--Flush Req----|<-State Req (STATUS_FLUSH_BUFS)--|<---Flush Req--

   ---Flush Cfm--->|--State Cfm (STATUS_FLUSH_BUFS)->|---Flush Cfm-->
```

The following message flow shows a request to continue.

```
  MTP2            M2UA                                 M2UA          MTP3
   SGP             SGP                                  ASP           ASP

   <---Cont Req----|<--State Req (STATUS_CONTINUE)---|<---Cont Req---

   ----Cont Cfm--->|---State Cfm (STATUS_CONTINUE)-->|----Cont Cfm-->
```

5.3.8 Auditing of SS7 link state

It may be necessary for the ASP to audit the current state of a SS7 link. The flows below show an example of the request and all the potential responses.

Below is an example in which the SS7 link is out-of-service.

```
 MTP2            M2UA                                 M2UA           MGMT
  SGP             SGP                                  ASP            ASP

                 |<----State Req (STATUS_AUDIT)----|<----Audit-------

                                                                     MTP3
                                                                      ASP

                 |-----------Release Ind---------->|-Out of Serv Ind->

                                                                     MGMT
                                                                      ASP

                 |-----State Cfm (STATUS_AUDIT)--->|----Audit Cfm--->
```

Below is an example in which the SS7 link is in-service.

```
 MTP2            M2UA                                 M2UA           MGMT
  SGP             SGP                                  ASP            ASP

                 |<----State Req (STATUS_AUDIT)----|<----Audit-------

                                                                     MTP3
                                                                      ASP

                 |-----------Establish Cfm-------->|---In Serv Ind-->

                                                                     MGMT
                                                                      ASP

                 |-----State Cfm (STATUS_AUDIT)--->|----Audit Cfm--->
```

Below is an example in which the SS7 link is in-service, but congested.

```
MTP2           M2UA                                  M2UA           MGMT
 SGP            SGP                                   ASP            ASP

                |<----State Req (STATUS_AUDIT)----|<----Audit-------

                                                                    MTP3
                                                                     ASP

                |-----------Establish Cfm-------->|---In Serv Ind-->

                |----------Congestion Ind-------->|---Cong Ind----->

                                                                    MGMT
                                                                     ASP

                |-----State Cfm (STATUS_AUDIT)--->|----Audit Cfm--->
```

Below is an example in which the SS7 link is in-service, but in Remote Processor Outage.

```
MTP2           M2UA                                  M2UA           MGMT
 SGP            SGP                                   ASP            ASP

                |<----State Req (STATUS_AUDIT)----|<---Audit Req----

                                                                    MTP3
                                                                     ASP

                |-----------Establish Ind-------->|---In Serv Ind-->

                |---State Ind (EVENT_RPO_ENTER)-->|----RPO Enter--->

                                                                    MGMT
                                                                     ASP

                |-----State Cfm (STATUS_AUDIT)--->|----Audit Cfm--->
```

6.0 Timer Values

The recommended default values for M2UA timers are:

T(r)		2 seconds
T(ack)		2 seconds
T(beat)	Heartbeat Timer	30 seconds

7.0 Security Considerations

M2UA is designed to carry signalling messages for telephony services. As such, M2UA MUST involve the security needs of several parties: the end users of the services; the network providers and the applications involved. Additional requirements MAY come from local regulation. While having some overlapping security needs, any security solution SHOULD fulfill all of the different parties' needs.

7.1 Threats

There is no quick fix, one-size-fits-all solution for security. As a transport protocol, M2UA has the following security objectives:

* Availability of reliable and timely user data transport.
* Integrity of user data transport.
* Confidentiality of user data.

M2UA runs on top of SCTP. SCTP [8] provides certain transport related security features, such as:

* Blind Denial of Service Attacks
* Flooding
* Masquerade
* Improper Monopolization of Services

When M2UA is running in a professionally managed corporate or service provider network, it is reasonable to expect that this network includes an appropriate security policy framework. The "Site Security Handbook" [13] SHOULD be consulted for guidance.

When the network in which M2UA runs in involves more than one party, it MAY NOT be reasonable to expect that all parties have implemented security in a sufficient manner. In such a case, it is recommended that IPSEC is used to ensure confidentiality of user payload. Consult [14] for more information on configuring IPSEC services.

7.2 Protecting Confidentiality

Particularly for mobile users, the requirement for confidentiality MAY include the masking of IP addresses and ports. In this case application level encryption is not sufficient; IPSEC ESP SHOULD be used instead. Regardless of which level performs the encryption, the IPSEC ISAKMP service SHOULD be used for key management.

8.0 IANA Considerations

8.1 SCTP Payload Protocol Identifier

A request will be made to IANA to assign an M2UA value for the Payload Protocol Identifier in SCTP Payload Data chunk. The following SCTP Payload Protocol Identifier has been registered:

M2UA "2"

The SCTP Payload Protocol Identifier is included in each SCTP Data chunk, to indicate which protocol the SCTP is carrying. This Payload Protocol Identifier is not directly used by SCTP but MAY be used by certain network entities to identify the type of information being carried in a Data chunk.

The User Adaptation peer MAY use the Payload Protocol Identifier as a way of determining additional information about the data being presented to it by SCTP.

8.2 M2UA Protocol Extensions

This protocol may also be extended through IANA in three ways:

- -- through definition of additional message classes,
- -- through definition of additional message types, and
- -- through definition of additional message parameters.

The definition and use of new message classes, types and parameters is an integral part of SIGTRAN adaptation layers. Thus, these extensions are assigned by IANA through an IETF Consensus action as defined in [RFC2434].

The proposed extension must in no way adversely affect the general working of the protocol.

8.2.1 IETF Defined Message Classes

The documentation for a new message class MUST include the following information:

(a) A long and short name for the message class.
(b) A detailed description of the purpose of the message class.

8.2.2 IETF Defined Message Types

Documentation of the message type MUST contain the following information:

(a) A long and short name for the new message type.
(b) A detailed description of the structure of the message.
(c) A detailed definition and description of intended use of each field within the message.
(d) A detailed procedural description of the use of the new message type within the operation of the protocol.
(e) A detailed description of error conditions when receiving this message type.

When an implementation receives a message type which it does not support, it MUST respond with an Error (ERR) message with an Error Code of Unsupported Message Type.

8.2.3 IETF-defined TLV Parameter Extension

Documentation of the message parameter MUST contain the following information:

(a) Name of the parameter type.
(b) Detailed description of the structure of the parameter field. This structure MUST conform to the general type-length-value format described in Section 3.1.5.
(c) Detailed definition of each component of the parameter value.
(d) Detailed description of the intended use of this parameter type, and an indication of whether and under what circumstances multiple instances of this parameter type may be found within the same message type.

9.0 Acknowledgments

The authors would like to thank Tom George (Alcatel) for contribution of text and effort on the specification.

The authors would like to thank John Loughney, Neil Olson, Michael Tuexen, Nikhil Jain, Steve Lorusso, Dan Brendes, Joe Keller, Heinz Prantner, Barry Nagelberg, Naoto Makinae, Joyce Archibald, Mark Kobine, Nitin Tomar, Harsh Bhondwe and Karen King for their valuable comments and suggestions.

10.0 References

10.1 Normative

[1] ITU-T Recommendation Q.700, 'Introduction To ITU-T Signalling System No. 7 (SS7)'

[2] ITU-T Recommendation Q.701-Q.705, 'Signalling System No. 7 (SS7) - Message Transfer Part (MTP)'

[3] ANSI T1.111 'Signalling System Number 7 - Message Transfer Part'

[4] Bellcore GR-246-CORE 'Bell Communications Research Specification of Signalling System Number 7', Volume 1, December 1995

[5] Telecommunication Technology Committee (TTC) Standard JT-Q704, Message Transfer Part Signaling Network Functions, April 28, 1992.

[6] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

[7] Coded Character Set--7-Bit American Standard Code for Information Interchange, ANSI X3.4-1986.

10.2 Informative

[8] Stewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Zhang, L. and V. Paxson, "Stream Control Transmission Protocol", RFC 2960, October 2000.

[9] Ong, L., Rytina, I., Garcia, M., Schwarzbauer, H., Coene, L., Lin, H., Juhasz, I., Holdrege, M. and C. Sharp, "Architectural Framework for Signalling Transport", RFC 2719, October 1999.

[10] ITU-T Recommendation Q.2140, 'B-ISDN ATM Adaptation Layer', February 1995

[11] ITU-T Recommendation Q.2210, 'Message transfer part level 3 functions and messages using the services of ITU-T Recommendation Q.2140', August 1995

[12] ITU-T Recommendation Q.751.1, 'Network Element Management Information Model for the Message Transfer Part', October 1995

[13] Fraser, B., "Site Security Handbook", FYI 8, RFC 2196, September 1997.

[14] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

Appendix A: Signalling Network Architecture

A Signalling Gateway will support the transport of MTP2-User signalling traffic received from the SS7 network to one or more distributed ASPs (e.g., MGCs). Clearly, the M2UA protocol description cannot in itself meet any performance and reliability requirements for such transport. A physical network architecture is required, with data on the availability and transfer performance of the physical nodes involved in any particular exchange of information. However, the M2UA protocol is flexible enough to allow its operation and management in a variety of physical configurations that will enable Network Operators to meet their performance and reliability requirements.

To meet the stringent SS7 signalling reliability and performance requirements for carrier grade networks, these Network Operators should ensure that there is no single point of failure provisioned in the end-to-end network architecture between an SS7 node and an IP ASP.

Depending of course on the reliability of the SGP and ASP functional elements, this can typically be met by spreading SS7 links in a SS7 linkset [1] across SGPs or SGs, the provision of redundant QoS-bounded IP network paths for SCTP Associations between SCTP End Points, and redundant Hosts. The distribution of ASPs within the available Hosts is also important. For a particular Application Server, the related ASPs MAY be distributed over at least two Hosts.

An example of logical network architecture relevant to carrier-grade operation in the IP network domain is shown in Figure 7 below:

```
     **************                              **************
     *  ********__*______________________________*__********  * Host1
SG1  *  * SGP1 *__*_________________      _______*__* ASP1 *  *
     *  ********  *                |     |       *  ********  *
     *      .     *                |     |       *            *
     *      .     *                |     |       **************
     **************                |     |
                                   |     |
     **************                |     |
     *  ********__*______________________|
SG2  *  * SGP2 *__*_________       |
     *  ********  *        |       |
     *      .     *        |       |
     *      .     *        |       |
     **************        |       |             **************
                           |       |_____________*__********  * Host2
                           |_____________________*__* ASP2 *  *
            .                                    *  ********  *
            .            SCTP Associations       *            *
            .                                    **************
                                                         .
                                                         .
                                                         .
```

Figure 7: Logical Model Example

To avoid a single point of failure, it is recommended that a minimum of two ASPs be configured in an AS list, resident in separate hosts and, therefore, available over different SCTP associations. For example, in the network shown in Figure 7, all messages for the Interface Identifiers could be sent to ASP1 in Host1 or ASP2 in Host2. The AS list at SGP1 might look like the following:

```
Interface Identifiers - Application Server #1
    ASP1/Host1  - State = Active
    ASP2/Host2  - State = Inactive
```

In this 1+1 redundancy case, ASP1 in Host1 would be sent any incoming message for the Interface Identifiers registered. ASP2 in Host2 would normally be brought to the active state upon failure of ASP1/Host1. In this example, both ASPs are Inactive or Active, meaning that the related SCTP association and far-end M2UA peer is ready.

For carrier grade networks, Operators should ensure that under failure or isolation of a particular ASP, stable calls or transactions are not lost. This implies that ASPs need, in some cases, to share the call/-transaction state or be able to pass the call/transaction state between each other. Also, in the case of ASPs performing call processing, coordination MAY be required with the related Media Gateway to transfer the MGC control for a particular trunk termination. However, this sharing or communication is outside the scope of this document.

11.0 Authors' Addresses

Ken Morneault
Cisco Systems Inc.
13615 Dulles Technology Drive
Herndon, VA. 20171
USA

Phone: +1-703-484-3323
EMail: kmorneau@cisco.com

Ram Dantu, Ph.D.
NetRake Corporation
3000 Technology Drive
Plano, TX 75074
USA

Phone: +1-214-291-1111
EMail: rdantu@netrake.com

Greg Sidebottom
Signatus Technologies
Kanata, Ontario, Canada

EMail: greg@signatustechnologies.com

Brian Bidulock
OpenSS7 Corporation
1469 Jeffreys Crescent
Edmonton, AB T6L 6T1
Canada

Phone: +1-780-490-1141
EMail: bidulock@openss7.org

Jacob Heitz
Lucent Technologies
1701 Harbor Bay Parkway
Alameda, CA, 94502
USA

Phone: +1-510-747-2917
EMail: jheitz@lucent.com

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.